

Почему программы должны быть свободны (Why Software Should Be Free)

Ричард Столлмен ([Richard Stallman](#))

Сергей Короп (пер. с англ.)

(Version of April 24, 1992)



Введение

Существование программного обеспечения неотвратимо порождает вопрос о том, как должны приниматься решения по использованию программ. Например, представьте, что некто, имеющий копию программы, встречает желающего скопировать ее себе. Такая возможность имеется, но кто должен разрешать сделать это? Двое участников? Или третья сторона, называемая ``собственником''?

Разработчики программ обычно рассматривают эти вопросы в предположении, что критерий правильного ответа --- максимизация их доходов. Политическое влияние бизнеса привело к тому, что правительство признало как этот критерий, так и ответ на вопрос, предложенный разработчиками: программа имеет владельца, обычно корпорацию, связанную с его разработкой.

Я хотел бы взглянуть на этот же вопрос под другим углом: процветание и свобода общества в целом.

Не стоит искать ответа на вопрос в сегодняшнем законодательстве---это законы должны соответствовать этике, а не наоборот. Точно так же не решает вопроса и существующая практика, хотя она в состоянии подсказать возможные ответы. Единственный путь обсуждения в том, чтобы понять, кому помогает, а кому вредит признание права

собственности на программы, и разобраться, почему это происходит и в какой мере. Другими словами, нам следует проанализировать затраты и прибыль в интересах общества в целом, принимая во внимание личную свободу в той же степени, что и производство материальных ценностей.

В этой статье я опишу последствия существования собственников и покажу, что результатом будет убыток. Мое заключение: программисты обязаны поощрять других делиться друг с другом, распространять, изучать и улучшать программы, которые мы создаем: другим словами, писать [``свободные" программы](#). (1)

Как собственники оправдывают свою власть

Те, кто извлекает выгоду из существующей системы собственности на программы, предлагают два аргумента в поддержку своих претензий на право собственности: эмоциональный аргумент и экономический.

Эмоциональный подобен следующему: ``Я трудился в поте лица, вложил в эту программу свою душу, свое сердце. Она появилась благодаря *мне*, она *моя*!"

Этот аргумент не требует усилий для опровержения. Чувство привязанности---одно из тех, которые программисты могут культивировать, когда им это удобно, оно не существует неизменно. Посмотрите, к примеру, как охотно те же самые программисты обычно уступают все права крупным корпорациям в обмен на заработную плату; эмоциональная привязанность таинственно исчезает. В отличие от этого, взглянем на великих художников и ремесленников средних веков, которые даже не подписывали своих работ. Для них имя художника было неважно. Имело значение, что работа была сделана, и цель, которой она должна была служить. Такая точка зрения преобладала на протяжении веков.

Экономический аргумент похож на такой: ``Я хочу разбогатеть (обычно некорректно подменяется `заработать на жизнь'), и если вы не позволите мне стать богатым при помощи программирования, я не буду программировать. Все остальные думают точно так же, поэтому никто не будет писать программы. И вы останетесь вообще без программ!" Эта угроза обычно замаскирована под дружеский совет благоразумных.

Позднее я поясню, почему эта угроза --- блеф. Сначала я хочу обратиться к неявному предположению, которое будет лучше видно, если сформулировать аргументы по-другому.

Иная формулировка начинается со сравнения социальной полезности собственнической программы и отсутствия программы вообще, далее

делается вывод, что в целом разработка собственнических программ полезна и должна поощряться. Посылка ошибочна, поскольку сравниваются лишь два возможных исхода --- собственнические программы против отсутствия программ --- и предположение, что других возможностей нет.

В данной системе интеллектуальной собственности разработка программ обычно связывается с существованием владельца, контролирующего их использование. Пока эта связь существует, мы зачастую вынуждены выбирать собственнические программы или никаких. Тем не менее, эта взаимосвязь не присуща изначально или неотвратима; она является следствием конкретного решения в области социально/правовой политики, которое мы обсуждаем: решения признать право собственности. Если сформулировать выбор, как необходимость или принять собственнические программы, или остаться ни с чем, то вопрос оказывается уже решенным.

Аргументы против наличия собственников

Вопрос, который нас интересует: ``Должна ли разработка программ связываться с наличием собственников, ограничивающих ее использование?"

Чтобы решить его, мы должны обсудить влияние на общество обоих видов деятельности *независимо*: эффект от разработки программ (независимо от условий их распространения), и эффект от ограничения его использования (подразумевается, что программы были разработаны). Если один из видов деятельности полезен, а другой приносит вред, было бы лучше отвергнуть их сцепление и впредь иметь дело только с полезным.

Говоря по-другому, если ограничение распространения программы, которая уже разработана, наносит вред всему обществу, то этический разработчик откажется от возможности так поступить.

Чтобы определить последствия ограничений на распространение, нам потребуется сравнить ценность для общества ограниченной (т.е. собственнической) программы и той же программы, доступной каждому. Это значит сравнить два возможных мира.

Этот анализ также касается применяемого иногда простого контраргумента, что ``выгода вашего друга, которому вы скопируете программу, обесценивается вредом, наносимым собственнику." Подобный аргумент предполагает, что выгода и ущерб одинакового порядка. В ходе анализа они сравниваются, и показывается, что выгода гораздо больше.

Чтобы разъяснить этот аргумент, давайте применим его к другой области:

прокладке дорог.

Возможно финансировать строительство всех дорог за счет сбора пошлин. Это может повлечь за собой строительство на каждом углу пунктов сбора пошлин. Такая система будет предоставлять прекрасный стимул к улучшению дорог. Она также выгодна тем, что пользователи каждой отдельно взятой дороги будут платить именно за нее. В то же время, строительство этих пунктов является искусственным ограничением для беспрепятственной езды,---искусственным потому, что не следует из принципа действия дороги либо автомобиля.

Сравнивая полезность свободных дорог с теми, на которых взимается пошлина, мы обнаруживаем, что при прочих равных дороги без шлагбаумов дешевле спроектировать и проложить, они безопаснее и эффективнее. (2) В бедной стране пошлины могут сделать дороги недоступными многим гражданам. Следовательно, беспошлинные дороги предлагают обществу большую выгоду при меньшей стоимости, они предпочтительны для общества. Поэтому обществу следует принять решение финансировать строительство дорог из других источников, не устанавливая пунктов сбора пошлин. Использование же дорог, однажды построенных, должно быть свободным.

Когда защитники сбора пошлины предлагают ее как *единственный* способ получения средств, они искажают доступные альтернативы. Сборщики пошлин приносят требуемые средства, но кроме того еще и другие последствия: фактически, они ведут ее к упадку. Дорога с пошлинами не так хороша, как свободная. Если нам предлагают большее количество дорог или техническое их усовершенствование, это не будет улучшением ситуации, если означает замену свободных дорог дорогами со сбором пошлин.

Безусловно, строительство свободной дороги требует денег, которые обществу придется каким-либо способом заплатить. Однако, это не подразумевает неотвратимости строительства пунктов сбора пошлин. Мы, кому придется платить так или иначе, получим больше за те же деньги, если купим себе свободную дорогу.

Я не утверждаю, что дорога, где взимают пошлины, хуже, чем отсутствие дорог вообще. Это было бы правдой, если бы пошлина была столь высокой, что не каждый смог бы пользоваться дорогой---но вряд-ли сборщики пошлин будут так поступать. В то же время, если пункты сбора пошлин вызывают значительные потери и неудобства, будет лучше получать требуемые средства в менее ограничительной манере.

Чтобы применить этот аргумент к разработке программного обеспечения,

я покажу, что ``обложение пошлиной'' полезного программного обеспечения дорого обходится обществу: создание и распространение программ оказываются более дорогостоящими, менее эффективным и приносит меньшее удовлетворение потребностей. Отсюда следует, что разработка программ должна стимулироваться иными путями. Далее я буду разъяснять другие методы поощрения и (в пределах действительной необходимости) финансирования разработки программ.

Вред, наносимый ограничениями на программы

Вообразите на минуту, что программа уже разработана, все необходимые для ее разработки выплаты уже сделаны, и теперь общество должно решить, сделать ли ее собственнической, или разрешить свободное распространение и использование. Предположим, что существование программы и ее доступность являются желательными. (3)

Ограничение распространения и модификации программы не может облегчить ее использования. Оно может лишь помешать. Следовательно, эффект может быть только негативным. Но насколько он велик? И какого рода?

Три вида материального ущерба следуют из таких препятствий:

- Меньшее количество людей воспользуется программой.
- Никто из пользователей не сможет адаптировать или исправить программу.
- Другие разработчики не смогут учиться на примере этой программы или использовать ее, как основу для новых работ.

Каждый уровень материального ущерба имеет сопутствующую форму психологического вреда. Под этим понимается эффект, оказываемый принимаемыми решениями на дальнейшие чувства, склонности и позиции людей. Эти перемены в способах размышления могут в будущем повлиять на их отношения с окружающими, а также иметь осязаемые последствия.

Упомянутые три уровня материального ущерба частично уменьшают приносимую программой пользу, но не снижают ее до нуля. Если они расходуют впустую почти всю ценность программы, то ее написание может повредить обществу в объеме, максимально равном затратам на разработку. Сомнительно, что программа, которую выгодно продавать, обязана принести некоторый чистый прямой материальный доход.

Тем не менее, принимая во внимание психологический ущерб, можно

считать, что вред, который может нанести собственническое ПО, ничем не ограничен.

Затруднение использования программ

Первый уровень вреда --- затруднение простого использования программы. Копия программы имеет предельную стоимость, близкую к нулевой (вы можете заплатить эту цену, сделав работу самостоятельно), так что в условиях свободного рынка ее цена должна быть близкой к нулю. Лицензионная плата значительно сдерживает использование программы. Если полезная многим программа является собственнической, гораздо меньшее количество людей сможет ней воспользоваться.

Легко показать, что общий взнос программы в общество понижается, если назначить ей владельца. Каждый потенциальный пользователь программы, оказавшись лицом к лицу с необходимостью платить, может принять решение заплатить, но может и отказаться использовать программу. Когда пользователь решает заплатить, происходит перенос материальных ценностей от одного участника сделки к другому, общая сумма выгоды не изменится. Но каждый раз, когда кто-то отказывается использовать программу, это наносит ему вред, не давая выгоды никому. Сумма нуля и отрицательного числа будет отрицательной.

Но трудозатраты, требуемые на *разработку* программы, не сокращаются. В результате эффективность процесса в целом, измеренная как удовлетворение нужд пользователей, доставленное часом работы, понижается.

Этим демонстрируется критическая разница между копиями программ и машин, стульев или сэндвичей. Машина для копирования материальных объектов существует пока что лишь на страницах научно-фантастических книг. Но программы копировать легко; каждый может произвести так много копий, как ему требуется, с очень небольшими усилиями. Это не так для материальных объектов, поскольку вещество из ниоткуда не возникает --- каждая копия должна быть получена из сырья тем же путем, что и оригинал.

В случае материальных объектов сдерживание их использования имеет смысл, поскольку для меньшего количества объектов потребуется меньше сырья и трудозатрат на их изготовление. Действительно, существуют начальные затраты и затраты на разработку, которые вкладываются в стоимость выпускаемой продукции. Но пока предельная стоимость продукции значительна, добавить к ней долю затрат на разработку не вносит существенной разницы. Кроме того, не требуется ограничивать

свободу обычных пользователей.

В то же время, навязывать цену на что-то, что в противном случае было бы бесплатно, создает значительные изменения. Устанавливаемая централизованно плата за распространение программы становится мощным сдерживающим фактором.

Более того, централизованное производство, практикуемое сегодня, неэффективно даже как способ доставки копий программ. Эта система включает укладку дисков и лент в ненужные упаковки, доставку больших партий по всему миру и складирование их перед продажей. Цена всего этого представляется как расходы на ведение бизнеса; в действительности это часть бесполезных затрат, вызванных существованием владельцев программ.

Разрушение сплоченности общества

Вообразите, что как вы, так и ваш друг сочли полезным использовать некую программу. Заботясь о его благе, вы должны ощущать, что правильное решение ситуации --- позволить вам обоим использовать ее. Предложение позволить воспользоваться программой лишь одному из вас, запрещая другому, вызовет разногласия: ни вы, ни ваш друг не сочтут его приемлемым.

Подписать типичную программную лицензию означает предать вашего друга: ``Я обещаю лишить своего друга этой программы, чтобы мне позволили самому иметь копию." Люди, делающие такой выбор, выдерживают внутреннюю психологическую борьбу, когда оправдывают себя, принижая важность помощи окружающим---этим наносится вред духу общества. Таков психологический ущерб, связанный с материальным вредом, который вызван затруднением использования программы.

Многие пользователи бессознательно распознают неправильность отказа от возможности поделиться с другими, а поэтому решают распространять программы, игнорируя лицензии и законы. Но зачастую они ощущают вину за такие поступки. Им известно, что придется нарушить закон, чтобы остаться хорошими друзьями, но при этом они продолжают признавать авторитет закона, и делают вывод, что быть хорошим другом (каким они есть) гадко и позорно. Это тоже разновидность психологического вреда, которого можно избежать, заранее решив, что подобные лицензии и законы аморальны.

Программисты также испытывают психологический ущерб, понимая, что результаты выполняемой работы многим пользователям будут недоступны.

Это ведет к тому, что они склоняются к цинизму и отрицанию.

Программист может с энтузиазмом описывать работу, которую находит технически восхитительной, но когда его спрашиваешь: ``Будет ли мне позволено воспользоваться ней?``, он сникает и соглашается, что ответом будет "нет". Чтобы избежать такой обескураженности, он либо игнорирует этот факт большую часть времени, либо применяет циничную аргументацию, скрывающую его важность.

Со времен Рейгана наибольшим дефицитом в США были не технические нововведения, а готовность сотрудничать на благо общества.

Бессмысленно поощрять первое за счет последнего.

Препятствия индивидуальной адаптации программ

Второй уровень материального ущерба --- отсутствие возможности доработать программу. Легкость модификации программ --- одно из их величайших преимуществ перед старыми технологиями. Но большая часть коммерческих продуктов недоступна для изменений, даже после того, как вы их купите. Вам доступно лишь взять их как есть, в виде "черного ящика", или отказаться --- и это все.

Программа, которую вы можете запускать, состоит из последовательности чисел, значение которых таинственно. Никто, даже хороший программист, не в состоянии легко изменить их, чтобы заставить программу делать нечто другое.

Программисты обычно работают с ``исходными текстами`` программы, которые написаны на языках программирования, таких как Fortran или C. В них применяются имена, чтобы определить используемые данные и фрагменты программы, а операции представляются символами, такими как `+` для сложения и `-` для вычитания. Языки разработаны для того, чтобы помочь программисту читать и изменять программы. Приведем в качестве примера программу вычисления расстояния между двумя точками на плоскости:

```
float
distance (p0, p1)
    struct point p0, p1;
{
    float xdist = p1.x - p0.x;
    float ydist = p1.y - p0.y;
    return sqrt (xdist * xdist + ydist * ydist);
}
```

Вот эта же программа в исполнимой форме, понимаемой компьютером, который я обычно использую:

1314258944	-232267772	-231844864	1634862
1411907592	-231844736	2159150	1420296208
-234880989	-234879837	-234879966	-232295424
1644167167	-3214848	1090581031	1962942495
572518958	-803143692	1314803317	

Исходные тексты полезны (по крайней мере, потенциально) каждому пользователю программы. Но большинству пользователей не разрешается иметь копий исходных текстов. Обычно исходники собственных программ сохраняются владельцами в секрете, чтобы никто другой не почерпнул чего-то из них. Пользователи получают только файлы с непонятными числами, которые будут исполняться компьютером. Это значит, что изменить программу сможет лишь ее владелец.

Однажды мне пришлось услышать рассказ коллеги о шести месяцах работы в банке, где создавалась программа, похожая на другую, которая уже была доступной на коммерческой основе. Она считала, что если бы удалось достать исходные тексты той коммерческой программы, ее легко можно было бы приспособить к своим нуждам. Банк даже был готов заплатить, но ему этого не позволили---исходные тексты были секретом. В результате ей пришлось в течение полугода выполнять работу, которая вошла в валовой национальный продукт, но в действительности была потраченной впустую.

Лаборатория искусственного интеллекта MIT (MIT Artificial Intelligence lab, AI lab) получила в 1977 году от Xerox в подарок графический принтер. Он управлялся свободными программами, которые мы дополнили многими удобными возможностями. К примеру, программа могла уведомить пользователя сразу же по завершении печати. В случае проблем, таких как отсутствие либо застревание бумаги, программа могла немедленно уведомить всех пользователей, у которых в очереди на печать стояли задания. Эти возможности способствовали гладкой работе.

Позднее Xerox предоставила AI lab новый, более быстрый принтер, один из первых лазерных принтеров. Его работу контролировали собственные программы, работающие на выделенном компьютере, так что мы не смогли добавить туда ни одной нашей любимой возможности. Мы могли приспособиться посылать уведомление, когда задание доставлялось на машину, управляющую принтером, но не тогда, когда оно было действительно напечатано (и задержка обычно была заметной). Не было способа определить, что задание уже напечатано, мы могли только гадать. И никому не сообщалось, что застряла бумага, так что принтер мог простаивать без обслуживания часами.

Системные программисты AI lab были в состоянии решить эти затруднения, возможно так же хорошо, как и авторы программы.

Компания Хегох не была заинтересована в решении проблем и решила оградить и нас от этого, так что нас заставили принять эти проблемы, как есть. Их так никогда и не исправили.

Большинство хороших программистов имеет опыт подобного разочарования. Банк мог попытаться выйти из положения, разработав новую программу с нуля, но обыкновенному пользователю, опытному или нет, остается только сдаться.

Когда вы сдаетесь, это также вызывает психологический вред---подрывает уверенность в себе. Жизнь в доме, который вы не можете обустроить под ваши нужды, деморализует. Она ведет к покорности и упадку духа, которые могут распространиться и на другие стороны жизни. Люди, испытывающие такие чувства, несчастливы и не могут хорошо работать.

Представьте себе, что было бы, если бы кулинарные рецепты были захвачены кем-то, подобно программам сегодня. Вы можете спросить: ``Как могу я изменить этот рецепт, чтобы в нем не было соли?``, а шеф-повар вам ответит: ``Как вы осмеливаетесь оскорблять мой рецепт, дитя моего ума и вкуса, пытаться вмешиваться в него? Вы не имеете права изменить мой рецепт и сделать его правильным!``

``Но врач не рекомендовал мне употреблять в пищу соль! Что я могу делать? Не могли бы вы для меня исключить соль?``

``Я рад сделать это, только заплатите \$50,000." Поскольку собственник располагает монополией на модификации, цена их имеет тенденцию быть высокой. ``Правда, именно сейчас у меня нет времени. Я занят в комиссии по созданию нового рецепта корабельного бисквита для военно-морских сил. Я смогу заняться вами где-то через пару лет."

Препятствия разработке программ

Третий уровень материального ущерба затрагивает разработку программ. Разработка программ часто является эволюционным процессом, когда разработчик может взять существующую программу и переписать ее часть, добавив еще одну новую возможность, а затем кто-либо другой может также переписать другую ее часть и добавить другую возможность; в некоторых случаях это продолжается более двадцати лет. В частности, фрагменты программы могут быть ``вырезаны", чтобы использоваться для создания новых программ.

Существование собственников препятствует такой эволюции, делая необходимым каждый раз начинать разработку программы с самого начала. Оно также препятствует новичкам изучать существующие

продукты, чтобы ознакомиться с полезными технологиями или даже понять, как может быть структурирована большая программа.

Собственники также создают препоны образованию. Я встречал блестящих студентов, изучающих компьютерные науки, которые никогда не видели исходных текстов большой программы. Они хорошо писали небольшие программы, но не могли начать учиться новым для себя навыкам создания крупных, если не могли увидеть, как это делают другие.

Во всех областях интеллектуальной деятельности можно достичь лучших результатов, стоя на плечах других. Но это, вообще говоря, более невозможно в области программирования---вы можете опираться лишь на людей, работающих *в вашей компании*.

Связанный с этим психологический вред затрагивает дух научного сотрудничества, который обычно бывает столь силен, что ученые могут сотрудничать, даже если их страны воюют между собой. Действуя в таком духе, японские океанографы, ликвидируя лабораторию на острове в Тихом океане, тщательно законсервировали свои работы, подготовив их к приходу занявших остров ВМС США, и оставили сообщение, попросив позаботиться о них.

Конфликт ради прибыли уничтожил то, что уцелело во время войны. Сегодня ученые многих специальностей не публикуют в своих статьях достаточно информации, чтобы другие могли воспроизвести эксперимент. Они ограничиваются объемом, достаточным, чтобы удивить слушателей, как много смогли сделать. Это, конечно, правда и для компьютерных наук, где исходные тексты программ, которым посвящаются публикации, обычно секретны.

Неважно, как ограничивается взаимообмен

Мною обсуждались последствия воспреещения копировать, изменять и дорабатывать программу. Я не конкретизировал, каким образом вводятся эти ограничения, поскольку это не влияет на сделанные выводы. Делаются ли они путем защиты от копирования или авторским правом, или лицензиями, или шифрованием, или электронными ключами, или серийными номерами аппаратуры, если они *эффективны в ограничении использования*, это приносит вред.

Пользователи считают некоторые из таких методов более неприятными, чем другие. Я полагаю, что те из методов, которые ненавидят сильнее всего, лучше других справляются со своим предназначением.

Программы должны быть свободными

Я показал, как право собственности на программу---полномочия ограничивать ее модификацию или копирование---является препятствием. Его негативные последствия широкомасштабны и важны. Отсюда следует, что обществу не следует иметь собственников программ.

Другой путь понять это таков, что в действительности общество нуждается в свободных программах, а собственническое ПО --- лишь слабая его замена. Поощрять заменитель не есть лучшим способом получить то, что нам требуется.

Вацлав Гавел советовал нам ``Делать нечто потому, что это правильно, а не только потому, что оно является шансом преуспеть." Предприятия по выпуску собственнических программ предлагают возможность успеха в своем собственном узком смысле, но это не то, что полезно для общества.

Почему люди будут разрабатывать программы

Если мы устраним интеллектуальную собственность как способ поощрять людей разрабатывать программы, то вначале будет разработано меньше программ, но они будут более полезны. Неочевидно, уменьшится ли суммарное удовлетворение, доставленное пользователям, но если уменьшится, либо мы пожелаем непременно увеличить его, существуют иные способы стимулировать разработку, подобно тому, как есть возможность финансировать дорожные работы помимо сбора пошлин. Перед тем, как обсудить возможные пути сделать это, я хотел бы спросить, насколько в действительности необходимо искусственное поощрение?

Программирование приносит радость

Существуют некоторые виды работ, которыми немногие согласятся заниматься, кроме как за деньги, например, прокладка дорог. Но есть и другие области науки и искусства, в которых шанс разбогатеть невелик, и которыми люди занимаются благодаря их привлекательности либо осознавая их важность для общества. Примерами могут служить математическая логика, классическая музыка, археология и политические организации среди рабочих. Люди состязаются, скорее уныло, чем жестоко, за небольшое количество оплачиваемых должностей, ни одна из которых не оплачивается очень хорошо. Они могут даже заплатить за шанс работать в некоторой области, если могут себе это позволить.

Такая область может измениться в одно мгновение, если начнет предлагать шанс разбогатеть. Когда один из работающих богатеет,

остальные потребуют той же возможности. Вскоре все могут потребовать крупные суммы денег за то, что ранее делали для своего удовольствия. Спустя еще пару лет каждый, кто связан с этой деятельностью, подвергнет насмешкам идею, что в данной области можно работать без крупных финансовых выплат. Они начнут советовать тем, кто выбирает дальнейший путь общества, обеспечить гарантированный возврат вложений, предписывая, что особые привилегии, власть и монополия для этого необходимы.

Эти перемены произошли в области программирования в последнее десятилетие. Пятнадцать лет назад выходили статьи о ``компьютеромании``: ``онлайновые`` пользователи ограничивались сотней долларов в неделю. Всем было понятно, что люди зачастую любят программирование достаточно сильно, чтобы разрушать свои браки. Сегодня общепринятым является мнение, что никто не будет программировать, кроме как за высокую плату. Люди позабыли то, что знали пятнадцать лет назад.

Даже когда в данное время правда то, что большинство людей будет работать в некоторой области лишь за высокую плату, все не обязательно останется именно так. Динамика перемен может повернуть вспять, если общество предложит стимул. Если мы устраним возможность крупно разбогатеть, то спустя какое-то время, когда люди пересмотрят свои позиции, они однажды снова будут готовы работать в этой области ради удовольствия от достижения совершенства.

Вопрос ``как мы сможем платить программистам?`` становится проще, когда мы понимаем, что это не обязательно должно быть целое состояние. Простую жизнь финансировать легче.

Финансирование свободного ПО

Организации, которые платят программистам, не обязательно должны быть фирмами по производству программ. Есть множество других, уже существующих, которые могли бы делать это.

Производители оборудования находят необходимым поддерживать разработку программ, даже если не могут контролировать их использование. В 1970 году большая часть их программ была свободной, поскольку они не рассматривали возможность ввести ограничения. Сегодня растущая готовность производителей оборудования вступать в консорциумы показывает их понимание, что владение программами --- не то, что действительно важно для них.

Университеты проводят множество программных проектов. Сегодня они частенько торгуют своими результатами, но в 1970-х такого не было. Сомневаетесь, что университеты будут разрабатывать свободные программы, если не смогут их продавать? Эти проекты могли бы поддерживаться теми же правительственными контрактами и грантами, которые сегодня питают разработку собственнических программ.

Сегодня обычна ситуация, когда университетские исследователи, чтобы получить грант на разработку системы, доводят ее практически до завершения и называют проект ``оконченным'', а впоследствии основывают компании, которые действительно заканчивают работу над ним и делают его полезным. Иногда они объявляют незавершенную версию ``свободной'', если же они окончательно испорчены, то вместо этого получают привилегированную лицензию от университета. Это не секрет, а открытое признание всех участников. Но если бы исследователи не поддавались искушению делать подобные вещи, они продолжали бы свои исследования.

Программисты-разработчики свободных программ могут заработать на жизнь, продавая услуги, связанные с этими программами. Мне приходилось работать по найму над переносом [компилятора GNU C](#) на новое оборудование и расширениями пользовательского интерфейса [GNU Emacs](#). (Я предложил эти улучшения обществу, когда закончил их.) Кроме этого я проводил учебу, за что мне платили.

На этом пути я не одинок, сегодня существуют успешные растущие корпорации, занимающиеся только такой работой. Некоторые другие компании также предлагают коммерческую поддержку свободных программ системы GNU. Зарождается индустрия независимой поддержки программ, которая может стать весьма обширной, если свободное ПО станет преобладать. Она предоставляет возможности, обычно недоступные пользователям собственнического ПО, за исключением очень богатых.

Новые организации, такие как [Фонд Свободного ПО](#) (Free Software Foundation) также могут финансировать программистов. Большая часть доходов фонда приносится пользователями, которые заказывают ленты по почте. Программы, записанные на эти ленты, свободные, а это значит, что каждый пользователь может свободно копировать и изменять их, но многие все равно платят, чтобы получить копию. (Помните, что ``свободные программы'' означают свободу, а не цену.) Некоторые пользователи заказывают ленты, уже располагая копией нужной программы, чтобы тем самым внести пожертвование, которого по их мнению мы заслужили. Фонд также получает изрядные пожертвования от производителей компьютеров.

Фонд --- благотворительная организация, и его доходы используются для найма такого количества программистов, которое будет возможно. Если бы он был основан как коммерческое предприятие по распространению тех же самых свободных программ той же публике за ту же плату, сегодня бы он обеспечил очень хорошую жизнь своему основателю.

Поскольку Фонд --- благотворительная организация, программисты зачастую работают для него за половину того, что они могли бы заработать где-либо еще. Они делают это, поскольку мы свободны от бюрократии, и потому, что ощущают удовлетворение, зная, что использованию их работы не будет препятствий. Большинство из них поступает так потому, что получает от программирования удовольствие. В дополнение к этому многие полезные программы были написаны для нас добровольцами. (С недавних пор даже технические писатели начали работать добровольцами.)

Этим подтверждается, что программирование располагается среди наиболее захватывающих видов деятельности, наряду с музыкой и живописью. Не стоит опасаться, что никто не захочет программировать.

Что пользователи должны разработчикам?

У пользователей свободных программ существует хороший повод ощущать моральный долг внести пожертвование на их поддержку. Разработчики свободных программ вносят вклад в деятельность пользователей, так что одновременно будет как честным, так и привлекательным для пользователей в долгосрочной перспективе предоставить им средства для продолжения работ.

В то же время, это не относится к авторам собственных программ, поскольку запретительство заслуживает скорее наказания, нежели награды.

Следовательно, мы пришли к парадоксу: разработчик полезных программ имеет право на поддержку со стороны пользователей, но любые попытки превратить чувство благодарности в обязанность разрушают основы этого чувства. Разработчик может либо заслуживать награды, либо требовать ее, но не одновременно.

Я верю, что этичный разработчик, оказавшись лицом к лицу с этим парадоксом, обязан поступать так, чтобы заслуживать награды, но должен также обращаться к пользователям с просьбой о добровольных пожертвованиях. Вскоре пользователи научатся поддерживать разработчиков без принуждения, подобно тому, как они уже привыкли делать это для общественного радио и телевизионных станций.

Что такое "программная продуктивность"?

Если программы будут свободны, будут существовать и программисты, но возможно, в меньшем количестве. Повредит ли это обществу?

Не обязательно. Сегодня в развитых странах меньше фермеров, чем в 1900 году, но мы не думаем, что это вредно для общества, поскольку эти немногие дают сейчас больше продуктов потребителям, чем многие в прошлом. Это называется повышением производительности труда. Свободные программы могут потребовать гораздо меньше программистов, чтобы удовлетворить имеющиеся потребности, поскольку программная продуктивность повышается на всех уровнях:

- Более широкое использование каждой разработанной программы.
- Возможность приспособить к своим нуждам существующие программы, вместо написания новых с самого начала.
- Улучшение образования программистов.
- Исключение дублирования разработок.

Те, кто возражает против сотрудничества, поскольку оно приведет к найму меньшего количества программистов, в действительности возражают против увеличения производительности труда. Несмотря на это, такие люди обычно согласны с широко распространенным убеждением, что программная индустрия требует повышения производительности. Как это может быть?

``Программная продуктивность" (software productivity) может означать две различные вещи: совокупная продуктивность разработки программ в целом, или же продуктивность индивидуальных проектов. Общая продуктивность есть то, что общество желало бы улучшить, и самый очевидный путь сделать это --- устранить искусственные препятствия сотрудничеству, которые эту продуктивность снижают. Но исследователи, которые изучают понятие ``программная продуктивность", фокусируются исключительно на втором, ограниченном, толковании продуктивности, улучшение которой требует сложных технологических усовершенствований.

Неизбежно ли состязание?

Неизбежно ли, что люди будут пытаться соревноваться друг с другом, пытаться превзойти соперников? Возможно, да. Но соревнование само по себе безвредно, вредит *драка*.

Существует множество видов конкуренции. Состязание может заключаться в попытках достичь большего, превзойти сделанное другими. Например, в старые добрые времена, мастера программирования состязались в том, кто сможет заставить компьютер делать наиболее удивительные вещи, либо кто напишет самую короткую или быструю программу для некоторой задачи. Такое состязание может принести пользу каждому, *до тех пор, пока* сохраняется спортивный дух.

Конструктивных состязаний достаточно, чтобы мотивировать великие достижения. Многие люди соревнуются в том, чтобы первыми посетить все страны мира, некоторые даже тратят на это целые состояния. Но они не подкупают капитанов, чтобы те высаживали конкурентов на необитаемых островах. Они довольствуются тем, что позволяют выиграть лучшему.

Соревнование перерастает в схватку, когда участники начинают попытки навредить другим, вместо того, чтобы совершенствовать себя --- когда девиз ``Пусть победит лучший" уступает место ``Победить должен я, лучший или нет." Собственнические программы вредны не потому, что являются разновидностью состязания, но потому, что представляют собой сражение между членами нашего общества.

Состязание в бизнесе не обязательно должно быть битвой. Например, когда конкурируют два бакалейных магазина, все их усилия направлены на улучшение собственной работы, а не на саботаж соперника. Но этим не демонстрируются особые обязательства бизнес-этики; скорее, в данной области бизнеса можно найти небольшое пространство для сражений, не переходя к физическому насилию. Не все области предпринимательства обладают сходными характеристиками. Утаивание информации, которая могла бы помочь каждому сделать шаг вперед, также является формой схватки.

Идеология бизнеса не готовит людей сопротивляться искушению перейти от состязания к битве. Некоторые формы войны с конкурентами были запрещены антимонопольным законодательством, законами о правдивой рекламе и т.п., но вместо того, чтобы обобщить эти запреты в принципиальный отказ от схваток с конкурентами, руководства компаний изобретают новые формы борьбы, которые не были особо запрещены. Ресурсы общества тратятся зря на экономический эквивалент гражданской войны.

``Почему Вы не уедете в Россию?"

В Соединенных Штатах каждый защитник взглядов, отличающихся от крайних форм популистского эгоизма, часто слышит это обвинение.

Например, оно выдвигалось против сторонников национальной системы здравоохранения, подобной тем, которые существуют во всех высокоразвитых странах. Оно направлялось против защитников общественной поддержки искусств, также общепринятой в прогрессивном мире. Идея, что граждане имеют какие-либо обязательства во имя общественного блага, в США связывается с коммунизмом. Но в какой мере эти идеи схожи?

Коммунизм, как он существовал в Советском Союзе, был системой тотального контроля, где регламентировалась всякая деятельность, предположительно для общего блага, но в действительности --- во имя членов Коммунистической Партии. Имевшееся же копировальное оборудование тщательно охранялось во избежание нелегального копирования.

Американская система интеллектуальной собственности практикует централизованный контроль за распространением программ и снабжает копировальную технику [автоматическими схемами защиты от копирования](#), чтобы воспретить нелегальное копирование.

В отличие от них, я работаю над созданием системы, в которой люди свободны в выборе своих собственных действий, в частности, могут свободно помогать своим друзьям, а также изменять и улучшать инструменты, используемые в повседневной жизни. Системы, основанной на добровольном сотрудничестве и децентрализации.

Следовательно, если мы обсуждаем взгляды по их сходству с советским общественным строем, то коммунистами скорее стоит называть собственников программ.

Вопрос о предпосылках

В этой работе я делаю предположение, что пользователь программы не менее важен, чем ее автор и даже его работодатель. Другими словами, их интересы и нужды имеют равный вес, когда мы решаем, какой путь наилучший.

Подобная предпосылка признается не всеми. Многие считают, что наниматель автора изначально гораздо важнее кого-либо еще. Они говорят, к примеру, что целью существования собственности на программы является предоставление этому нанимателю преимуществ, которых он заслуживает, и неважно, как это может затрагивать общество.

Бесполезно пытаться доказать или опровергнуть подобные аргументы. Доказательство требует общепризнанных предпосылок. Поэтому большая

часть того, что я должен сказать, адресована тем, кто разделяет мои послышки, или по крайней мере заинтересованы в том, какие из них можно сделать выводы. Для тех, кто верит, что собственники важнее кого-либо еще, эта статья попросту не подходит.

Но почему большое количество американцев должно принимать предположение, которое возвышает значимость некоторых людей над всеми остальными? Частично благодаря уверенности, что это предположение является частью правовых традиций американского общества. Некоторые ощущают, что сомневаться в нем означает бросать вызов основам общества.

Таким людям важно знать, что эта предпосылка не является частью наших правовых традиций и никогда ней не была.

Так, Конституция США утверждает, что назначение системы авторских прав в том, чтобы ``способствовать прогрессу науки и полезных искусств." Верховный Суд развил эту идею, постановив в деле Fox Film Corp. против Doyal, что: "Единственный интерес Соединенных Штатов и основная причина наделить кого-либо монопольным авторским правом лежит в пользе, получаемой обществом из труда авторов" ("The sole interest of the United States and the primary object in conferring the [copyright] monopoly lie in the general benefits derived by the public from the labors of authors").

Мы не обязаны во всем соглашаться с Конституцией или Верховным Судом (в свое время они смотрели сквозь пальцы на существование рабства.) Поэтому их позиции не опровергают предпосылки о превосходстве владельцев. Но я надеюсь, понимание того, что эта посылка --- радикальное реакционное предположение, а не общепризнанная традиция, ослабит ее притягательность.

Заключение

Нам приятно думать, что наше общество поощряет помощь вашим соседям, но каждый раз, когда мы награждаем кого-либо за обструкционизм или восхищаемся их богатством, добытым этим путем, мы утверждаем обратное.

"Захват" программ --- одна из форм нашей общей готовности пренебречь благом общества во имя личной выгоды. Мы можем проследить это пренебрежение от Рональда Рейгана (Ronald Reagan) до Джима Беккера (James Baker), от Ivan Boesky до Еххон, от лопающихся банков до закрывающихся школ. Мы можем измерить его количеством бездомных и преступников. Антисоциальный дух питает сам себя, поскольку чем больше мы видим, что другие нам не помогут, тем более бесполезным

кажется помогать им. Такое общество катится прямиком в джунгли.

Если мы не хотим жить в джунглях, мы должны изменить нашу позицию. Мы можем начать с распространения идеи, что хороший гражданин тот, кто сотрудничает когда это необходимо, а не тот, кто преуспел в искусстве обдирать других. Я надеюсь, что движение за свободное ПО внесет свою лепту: по крайней мере в одной области человеческой деятельности мы сможем заменить закон джунглей более эффективной системой, которая поощряет и основана на добровольном сотрудничестве.

Примечания

1) Слово ``свободный" (``free") в понятии ``свободные программы" (``free software") подразумевает свободу, а не цену. Цена, выплаченная за копию свободной программы, может быть нулевой или небольшой или (изредка) довольно крупной.

2) Если рассмотреть возможность загрязнения и возникновения пробок, то заключения это не изменит. Если мы хотим сделать вождение автомобиля более дорогим, чтобы затруднить его вообще, нерационально делать это с помощью пунктов сбора пошлин, поскольку увеличится загрязнение и возникнут заторы. Гораздо лучше ввести налог на бензин. Аналогично, желание увеличить безопасность, ограничивая скорость движения тоже неподходяще: дороги со свободным доступом улучшают среднюю скорость, устраняя необходимость остановок и задержек, независимо от предельной скорости.

3) Некто может обратить внимание, что существуют вредоносные программы, которые не должны быть доступны вообще, подобно базе личной информации Lotus Marketplace, которая была изъята из продажи вследствие неодобрения публики. Большая часть из сказанного мной не применима в этом случае, и возникает некоторый аргумент в пользу существования собственника на той основе, что он сделает эту программу менее доступной. Собственник не сделает ее *полностью* недоступной, как можно было бы желать в случае программы, использование которой считается деструктивным.

Другие тексты

Возврат к [титuleйной странице GNU](#) (Англ.).

Вопросы о деятельности FSF и проекте GNU направляйте по адресу gnu@gnu.org, либо свяжитесь с FSF [иным способом](#).

Внимание! Эта страница НЕ поддерживается FSF, который не несет никакой ответственности за ее содержание и/или оформление.

Бета-версия! Эта предварительная версия перевода может в дальнейшем подвергаться изменениям. Если Вы заинтересованы в ее публикации, сообщите об этом, и я вышлю Вам по окончании работы готовую версию. Если у Вас есть конструктивные замечания по стилю и качеству перевода---[пишите](#).

Copyright (C) 1998 Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110, USA

© 2000 Перевод на русский язык: Сергей Короп [<svk@lib.ru>](mailto:svk@lib.ru).

Разрешается копирование и распространение этой статьи любым способом без внесения изменений, при условии, что это разрешение сохраняется.

Verbatim copying and distribution of this entire article is permitted in any medium, provided this notice is preserved.

Перевод выполнен по версии статьи от 23 апреля 2000.
