# Почему вы должны использовать лицензию в стиле BSD для вашего Open Source проекта

## Содержание


1. Введение . . . 1
2. Очень краткая история открытого исходного кода . . . 1
3. Unix с точки зрения лицензирования BSD . . . 2
4. Текущее состояние FreeBSD и лицензий BSD . . . 3
5. Истоки GPL . . . 3
6. Истоки Linux и LGPL . . . 5
7. Открытые лицензии и проблема заброшенности проектов . . . 5
8. Что лицензия не может сделать . . . 6
9. Преимущества и недостатки GPL . . . 6
10. Преимущества BSD . . . 7
11. Конкретные рекомендации по использованию лицензии BSD . . . 8
12. Заключение . . . 10
Библиографические ссылки . . . 10


# 1. Введение

Этот документ обосновывает использование лицензии в стиле BSD для программного обеспечения и данных; в частности, он рекомендует применять лицензию в стиле BSD вместо GPL. Его также можно рассматривать как введение и краткое сравнение открытых лицензий BSD и GPL.

# 2. Очень краткая история открытого исходного кода

Долгое время до того, как термин "Открытое программное обеспечение" вошел в употребление, программное обеспечение разрабатывалось свободными объединениями программистов и свободно обменивалось. Начиная с ранних 1950-х годов, организации, такие как SHARE и DECUS, разрабатывали большую часть программного обеспечения, которое компании-производители аппаратного обеспечения поставляли вместе со своими продуктами. В то время компьютерные компании занимались аппаратным обеспечением;

все, что снижало стоимость программного обеспечения и делало больше программ доступными, повышало конкурентоспособность производителей аппаратного обеспечения.

Эта модель изменилась в 1960-х годах. В 1965 году ADR разработала первый лицензированный программный продукт, независимый от компании-производителя оборудования. ADR конкурировала с бесплатным пакетом IBM, изначально разработанным клиентами IBM. ADR запатентовала своё программное обеспечение в 1968 году. Чтобы предотвратить распространение своей программы, они предоставляли её по договору аренды оборудования, где оплата распределялась на весь срок службы продукта. Таким образом, ADR сохраняла право собственности и могла контролировать перепродажу и повторное использование.

В 1969 году Министерство юстиции США обвинило IBM в уничтожении бизнесов путем объединения бесплатного программного обеспечения с аппаратным обеспечением IBM. В результате этого иска IBM прекратила практику объединения; то есть, программное обеспечение стало независимым продуктом, отдельным от аппаратного обеспечения.

В 1968 году компания Informatics представила приложение — первый коммерческий хит продукт, и быстро утвердила концепцию программного продукта, программной компании и очень высоких норм прибыли. Informatics разработала бессрочную лицензию, которая теперь является стандартом во всей компьютерной отрасли, при которой право собственности никогда не передаётся клиенту.

# 3. Unix с точки зрения лицензирования BSD

AT&T, владевшая оригинальной реализацией Unix, была регулируемой государством монополией, связанной антимонопольными судебными разбирательствами; юридически она не могла продавать продукт на рынке программного обеспечения. Однако она имела возможность предоставлять его учебным заведениям по цене носителя.

Университеты быстро приняли Unix после того, как конференция по операционным системам объявила о его доступности. Крайне полезным было то, что Unix работал на PDP-11 — очень доступном 16-битном компьютере — и был написан на языке высокого уровня, который явно подходил для системного программирования. У DEC PDP-11, по сути, был открытый аппаратный интерфейс, разработанный для того, чтобы клиентам было легко писать свою собственную ОС, что было обычной практикой. Как знаменито заявил основатель DEC Кен Олсен: «программное обеспечение приходит с небес, когда у тебя хорошее железо».

Автор Unix Кен Томпсон вернулся в свой альма-матер, Калифорнийский университет в Беркли (UCB), в 1975 году и преподавал, построчно рассказывая, как работает ядро. В итоге это привело к созданию развивающейся системы, известной как BSD (Berkeley Standard Distribution). UCB перевел Unix на 32-битную архитектуру, добавил виртуальную память и реализовал версию стека TCP/IP, на котором, по сути, был построен Интернет. UCB сделал BSD доступным за стоимость носителя по условиям лицензии, которая стала известна как "лицензия BSD". Клиент покупал Unix у AT&T, а затем заказывал ленту BSD у UCB.

В середине 1980-х годов антимонопольный иск правительства против AT&T завершился разделом компании. AT&T по-прежнему владела Unix и теперь могла продавать его. AT&T начала агрессивную кампанию по лицензированию, и большинство коммерческих Unix того времени стали производными от AT&T.

В начале 1990-х годов AT&T подала в суд на UCB за нарушения лицензий, связанных с BSD. UCB обнаружила, что AT&T включила в свои продукты множество улучшений, сделанных в BSD, без указания авторства или оплаты, и последовало длительное судебное разбирательство, в основном между AT&T и UCB. В этот период некоторые программисты UCB начали проект по переписыванию любого кода AT&T, связанного с BSD. В результате этого проекта появилась система под названием BSD 4.4-lite (lite, потому что это была не полная система; в ней отсутствовали 6 ключевых файлов AT&T).

Длинная серия статей, опубликованных чуть позже в журнале Dr. Dobbs, описывала версию Unix для ПК на базе 386, производную от BSD, с файлами-заменителями под лицензией BSD для 6 отсутствующих файлов из 4.4 lite. Эта система, названная 386BSD, была создана бывшим программистом из UCB Уильямом Джолитцем. Она стала первоосновой всех современных PC BSD.

В середине 1990-х годов Novell приобрела права на Unix у AT&T, и было достигнуто (тогда секретное) соглашение о прекращении судебного разбирательства. Вскоре UCB прекратила поддержку BSD.

# 4. Текущее состояние FreeBSD и лицензий BSD

Так называемая новая лицензия BSD, применяемая к FreeBSD в последние годы, по сути, означает, что вы можете делать что угодно с программой или её исходным кодом, но не получаете никаких гарантий, и ни один из авторов не несёт ответственности (по сути, вы не можете ни на кого подать в суд). Эта новая лицензия BSD призвана стимулировать коммерциализацию продукта. Любой код BSD может быть продан или включён в проприетарные продукты без каких-либо ограничений на доступность вашего кода или ваше будущее поведение.

Не путайте новую лицензию BSD с "общественным достоянием" (лицензия Public Domain). Хотя объект в общественном достоянии также свободен для использования всеми, у него нет владельца.

# 5. Истоки GPL

Пока будущее Unix оставалось неясным в конце 1980-х и начале 1990-х годов, GPL, ещё одна разработка с важными лицензионными аспектами, достигла зрелости.

Ричард Столлман, разработчик Emacs, был сотрудником MIT, когда его лаборатория перешла с собственных разработок на проприетарные системы. Столлман был расстроен, обнаружив, что не может законно вносить даже незначительные улучшения в систему. (Многие его коллеги ушли, чтобы основать две компании, основанные на программном

обеспечении, разработанном в MIT и лицензированном MIT; по-видимому, возникли разногласия по поводу доступа к исходному коду этого программного обеспечения). Столлман разработал альтернативу коммерческой лицензии на программное обеспечение и назвал её GPL, или «Универсальная общественная лицензия GNU» (GNU Public License). Он также основал некоммерческую организацию — Фонд свободного программного обеспечения (FSF), целью которой была разработка целой операционной системы, включая все сопутствующие программы, которая не подчинялась бы проприетарным лицензиям. Эта система была названа GNU, что означает «GNU is Not Unix» (GNU — не Unix).

GPL была создана как антитеза стандартной проприетарной лицензии. Для этого любые изменения, внесённые в программу под GPL, должны были возвращаться сообществу GPL (путем требования, чтобы исходный код программы был доступен пользователю), и любая программа, использующая или связывающаяся с кодом под GPL, должна была распространяться под GPL. Лицензия GPL предназначена для предотвращения перехода программного обеспечения в проприетарное состояние. Как сказано в последнем абзаце GPL:

"This General Public License does not permit incorporating your program into proprietary programs (Стандартная Общественная Лицензия GNU не разрешает включать вашу программу в программы, использование которых ограничено их правообладателями)."[1]

GPL — это сложная лицензия, поэтому вот несколько эмпирических правил при использовании GPL:

- вы можете брать сколько угодно за распространение, поддержку или документацию к программному обеспечению, но вы не можете продавать само программное обеспечение.
- эмпирическое правило гласит, что если для компиляции программы требуется исходный код под GPL, то программа должна распространяться под лицензией GPL. Статическая линковка с библиотекой под GPL требует, чтобы программа также была под GPL.
- В соответствии с GPL, любые патенты, связанные с ПО под лицензией GPL, должны быть лицензированы для свободного использования всеми.
- простое объединение программного обеспечения, например, размещение нескольких программ на одном диске, не считается включением программ под лицензией GPL в программы, не подпадающие под GPL.
- вывод программы не считается производным произведением. Это позволяет использовать компилятор gcc в коммерческих средах без юридических проблем.
- поскольку ядро Linux распространяется под лицензией GPL, любой код, статически связанный с ядром Linux, должен быть лицензирован под GPL. Это требование можно обойти, используя динамическую загрузку модулей ядра. Это позволяет компаниям распространять двоичные драйверы, но часто имеет недостаток в том, что они будут работать только с определёнными версиями ядра Linux.

Из-за своей сложности во многих частях мира сегодня юридические аспекты GPL игнорируются в отношении Linux и связанного с ним программного обеспечения. Долгосрочные последствия этого неясны.

# 6. Истоки Linux и LGPL

Пока бушевали коммерческие войны Unix, ядро Linux разрабатывалось как клон Unix для ПК. Линус Торвальдс признаёт, что существование компилятора GNU C и связанных с ним инструментов GNU стало основой для появления Linux. Он выпустил ядро Linux под лицензией GPL.

Помните, что лицензия GPL требует, чтобы любой код, статически связанный с кодом под GPL, также распространялся под GPL. Исходный код такой программы должен быть предоставлен пользователю. Однако динамическая линковка не считается нарушением GPL. Давление с целью размещения проприетарных приложений в Linux стало слишком сильным. Такие приложения часто должны быть связаны с системными библиотеками. Это привело к созданию модифицированной версии GPL под названием LGPL ("Library", позже переименована в "Lesser" GPL). LGPL разрешает проприетарному коду быть связанным с библиотекой GNU C, glibc. Вам не нужно раскрывать исходный код, который был динамически связан с библиотекой под LGPL.

Если вы статически связываете приложение с glibc, как это часто требуется во встроенных системах, вы не можете сохранить ваше приложение проприетарным, то есть исходный код должен быть опубликован. И GPL, и LGPL требуют, чтобы любые изменения кода, подпадающего под лицензию, были выпущены.

# 7. Открытые лицензии и проблема заброшенности проектов

Одной из серьёзных проблем, связанных с проприетарным программным обеспечением, является так называемое "осиротение". Это происходит, когда единичный сбой в бизнесе или изменение стратегии продукта приводит к краху огромной пирамиды зависимых систем и компаний по причинам, не зависящим от них. Десятилетия опыта показали, что текущий размер или успех поставщика программного обеспечения не гарантирует, что их ПО останется доступным, поскольку рыночные условия и стратегии могут быстро меняться.

Лицензия GPL пытается предотвратить потерю поддержки, разрывая связь с проприетарной интеллектуальной собственностью.

Лицензия BSD предоставляет небольшой компании эквивалент программного обеспечения на условном депонировании без каких-либо юридических сложностей или затрат. Если программа под лицензией BSD становится заброшенной, компания может просто взять её под свой контроль в проприетарном режиме, если она от неё зависит. Ещё лучше, когда кодовая база BSD поддерживается небольшим неформальным консорциумом, поскольку процесс разработки не зависит от выживания отдельной компании или линейки продуктов. Выживаемость команды разработчиков, когда они находятся в режиме продуктивной работы, гораздо важнее простой физической доступности исходного кода.

# 8. Что лицензия не может сделать

Ни одна лицензия не может гарантировать доступность программного обеспечения в будущем. Хотя владелец авторских прав традиционно может изменять условия авторского права в любое время, в сообществе BSD принято считать, что такая попытка просто приведёт к ответвлению исходного кода.

Лицензия GPL явно запрещает отзыв лицензии. Однако случалось, что компания (Mattel) приобрела авторские права на продукт под GPL (cphack), отозвала все авторские права, обратилась в суд и выиграла дело [2]. То есть они легально отозвали всё распространение и все производные работы, основанные на этих авторских правах. Остаётся открытым вопрос, может ли подобное произойти с более крупным и распределённым проектом; также есть некоторая путаница относительно того, действительно ли данное ПО распространялось под лицензией GPL.

В другом примере, Red Hat приобрела Cygnus, инженерную компанию, которая взяла на себя разработку инструментов компилятора FSF. Cygnus смогла это сделать, потому что разработала бизнес-модель, в которой они продавали поддержку для программного обеспечения GNU. Это позволило им нанять около 50 инженеров и направлять развитие программ, внося основную часть изменений в код программы. Как отмечает Дональд Розенберг: «проекты, использующие лицензии вроде GPL… живут под постоянной угрозой, что кто-то перехватит проект, создав лучшую версию кода и сделав это быстрее, чем оригинальные владельцы.» [3]

# 9. Преимущества и недостатки GPL

Распространённая причина использовать GPL — это модификация или расширение компилятора gcc. Это особенно актуально при работе с уникальными специализированными процессорами в средах, где все затраты на программное обеспечение, скорее всего, считаются накладными расходами, с минимальными ожиданиями того, что другие будут использовать получившийся компилятор.

Лицензия GPL также привлекательна для небольших компаний, продающих CD в условиях, где принцип «купи-дешево, продай-дорого» может по-прежнему обеспечить конечному пользователю очень недорогой продукт. Она также привлекательна для компаний, которые рассчитывают выжить за счет предоставления различных форм технической поддержки, включая документацию, для мира интеллектуальной собственности под GPL.

Менее известное и непреднамеренное использование GPL заключается в том, что она очень выгодна крупным компаниям, которые хотят подорвать позиции компаний-разработчиков программного обеспечения. Другими словами, GPL хорошо подходит для использования в качестве маркетингового оружия, потенциально снижая общую экономическую выгоду и способствуя монополистическому поведению.

Лицензия GPL может представлять реальную проблему для тех, кто хочет коммерциализировать программное обеспечение и получать от него прибыль. Например, GPL усложняет задачу выпускника, который хочет создать компанию для коммерциализации результатов своих исследований, или затрудняет ситуацию для

студента, который планирует присоединиться к компании в расчете на коммерциализацию перспективного исследовательского проекта.

Для тех, кто должен работать со статически связанными реализациями множества программных стандартов, GPL часто является неудобной лицензией, поскольку она исключает использование проприетарных реализаций этих стандартов. Таким образом, GPL минимизирует количество программ, которые могут быть созданы с использованием стандарта под лицензией GPL. Лицензия GPL задумывалась так, чтобы не предоставлять механизм для разработки стандарта, на основе которого создаются проприетарные продукты. (Это не относится к приложениям для Linux, поскольку они не используют статическую линковку, а вместо этого применяют API на основе трапов/ловушек.)

Лицензия GPL пытается заставить программистов вносить вклад в развивающийся набор программ, а затем конкурировать в распространении и поддержке этого набора. Такая ситуация нереалистична для многих необходимых стандартов ядра системы, которые могут применяться в самых разных средах, требующих коммерческой настройки или интеграции с унаследованными стандартами под существующими (не-GPL) лицензиями. Системы реального времени часто статически связываются, поэтому GPL и LGPL определённо рассматриваются многими компаниями, работающими с встраиваемыми системами, как потенциальные проблемы.

GPL - это попытка удержать усилия, независимо от спроса, на этапах исследований и разработки. Это максимизирует выгоды для исследователей и разработчиков, при неизвестных затратах для тех, кто мог бы получить выгоду от более широкого распространения.

GPL была разработана для предотвращения перехода результатов исследований в проприетарные продукты. Этот шаг часто считается последним этапом в традиционном процессе передачи технологий, и он обычно достаточно сложен даже в самых благоприятных условиях; GPL была призвана сделать это невозможным.

# 10. Преимущества BSD

Лицензия в стиле BSD — это хороший выбор для долгосрочных исследований или других проектов, которым требуется среда разработки, обладающая следующими характеристиками:

- имеет почти нулевую стоимость
- будет развиваться в течение длительного времени
- позволяет любому сохранить возможность коммерциализации конечных результатов с минимальными юридическими проблемами.

Это последнее соображение часто может быть решающим, как это было, когда проект Apache выбирал свою лицензию:

"Этот тип лицензии идеально подходит для продвижения использования эталонного кода, реализующего протокол для общего сервиса. Это ещё одна причина, по которой мы выбрали его для группы Apache - многие из нас хотели, чтобы HTTP выжил и стал по-настоящему

многосторонним стандартом, и мы нисколько не возражали бы, если бы Microsoft или Netscape решили включить наш HTTP-движок или любой другой компонент нашего кода в свои продукты, если бы это способствовало достижению цели сохранения HTTP общим… Всё это означает, что, стратегически говоря, проект должен сохранять достаточную динамику, а участники должны осознавать большую ценность внесения своего кода в проект, даже того кода, который мог бы иметь ценность, оставаясь проприетарным."

Разработчики часто находят лицензию BSD привлекательной, так как она минимизирует юридические сложности и позволяет им делать с кодом всё, что угодно. В отличие от этого, те, кто в первую очередь планирует использовать систему, а не программировать её, или ожидает, что другие будут развивать код, или не рассчитывает зарабатывать на жизнь работой, связанной с системой (например, государственные служащие), находят GPL привлекательной, потому что она обязывает других разработчиков предоставлять им код и не позволяет их работодателю сохранять авторские права, что потенциально может привести к «забвению» или потере поддержки программного обеспечения. Если вы хотите заставить своих конкурентов помогать вам, GPL привлекательна.

Лицензия BSD — это не просто подарок. Вопрос «почему мы должны помогать нашим конкурентам или позволять им красть нашу работу?» часто возникает в связи с лицензией BSD. Под лицензией BSD, если одна компания начинает доминировать в нише продукта, который другие считают стратегическим, остальные компании могут с минимальными усилиями создать мини-консорциум, направленный на восстановление паритета путем внесения вклада в конкурирующий вариант BSD, что увеличивает конкуренцию и справедливость на рынке. Это позволяет каждой компании верить, что она сможет извлечь прибыль из какого-либо преимущества, которое она может предоставить, одновременно способствуя экономической гибкости и эффективности. Чем быстрее и проще сотрудничающие участники смогут это сделать, тем успешнее они будут. Лицензия BSD по сути является минимально сложной лицензией, которая делает такое поведение возможным.

Ключевой эффект GPL, делающий полную и конкурентоспособную систему с открытым исходным кодом широко доступной по стоимости носителя, является разумной целью. Лицензия в стиле BSD, в сочетании с ad-hoc-консорциумами индивидуумов, может достичь этой цели без разрушения экономических предположений, заложенных в развертывающем конце трубопровода передачи технологий.

# 11. Конкретные рекомендации по использованию лицензии BSD

- Лицензия BSD предпочтительна для передачи результатов исследований таким образом, чтобы они широко внедрялись и приносили максимальную пользу экономике. Таким образом, агентства, финансирующие исследования, такие как NSF, ONR и DARPA, должны на самых ранних этапах финансируемых исследовательских проектов поощрять принятие лицензий в стиле BSD для программного обеспечения, данных, результатов и открытого аппаратного обеспечения. Они также должны поощрять создание стандартов на основе реализованных систем с открытым исходным кодом и текущих проектов с открытым исходным кодом.

- Политика правительства должна минимизировать затраты и сложности перехода от исследований к внедрению. По возможности, гранты должны требовать, чтобы результаты были доступны под дружественной к коммерциализации лицензией в стиле BSD.
- Во многих случаях долгосрочные результаты лицензии в стиле BSD более точно отражают цели, провозглашенные в исследовательском уставе университетов, чем когда результаты защищены авторским правом или запатентованы и подлежат проприетарному лицензированию университета. Существуют неподтвержденные данные, что в долгосрочной перспективе университеты получают больше финансовых выгод, публикуя результаты исследований и затем обращаясь за пожертвованиями к коммерчески успешным выпускникам.
- Компании давно осознают, что создание де-факто стандартов является ключевым маркетинговым приемом. Лицензия BSD хорошо подходит для этой роли, если компания действительно обладает уникальным преимуществом в развитии системы. Лицензия юридически привлекательна для широкой аудитории, в то время как экспертиза компании гарантирует их контроль. Бывают случаи, когда GPL может быть подходящим инструментом для попытки создать такой стандарт, особенно при попытке подорвать или кооптировать другие. Однако GPL наказывает за эволюцию этого стандарта, потому что она продвигает набор инструментов, а не коммерчески применимый стандарт. Использование такого набора постоянно поднимает вопросы коммерциализации и юридические проблемы. Может оказаться невозможным совмещать стандарты, если некоторые из них под GPL, а другие — нет. Настоящий технический стандарт не должен требовать исключения других стандартов по нетехническим причинам.
- Компании, заинтересованные в продвижении развивающегося стандарта, который может стать основой коммерческих продуктов других компаний, должны остерегаться GPL. Независимо от используемой лицензии, итоговое программное обеспечение обычно переходит к тем, кто фактически вносит большинство инженерных изменений и лучше всего понимает состояние системы. GPL просто добавляет больше юридических сложностей к результату.
- Крупные компании, занимающиеся разработкой открытого исходного кода, должны понимать, что программисты ценят Open Source, потому что это позволяет сохранить доступ к программному обеспечению при смене работодателя. Некоторые компании поощряют такое поведение как дополнительное преимущество работы, особенно когда задействованное ПО не имеет прямого стратегического значения. По сути, это предварительно предоставленная пенсионная льгота с потенциальными упущенными возможностями, но без прямых затрат. Поощрение сотрудников к работе на признание коллег вне компании — это недорогое и переносимое преимущество, которое компания иногда может предоставить практически без негативных последствий.
- Небольшие компании с программными проектами, уязвимыми к заброшенности, должны по возможности использовать лицензию BSD. Компании любого размера должны рассмотреть возможность создания подобных проектов с открытым исходным кодом, когда это взаимовыгодно для поддержания минимальных юридических и организационных накладных расходов, связанных с настоящим проектом в стиле BSD с открытым исходным кодом.
- Некоммерческие организации должны по возможности участвовать в проектах с

открытым исходным кодом. Чтобы минимизировать проблемы в разработке программного обеспечения, такие как смешивание кода под разными лицензиями, следует поощрять лицензии в стиле BSD. Осторожность в отношении GPL особенно важна для некоммерческих организаций, взаимодействующих с развивающимися странами. В некоторых регионах, где применение закона становится дорогостоящим процессом, простота новой лицензии BSD по сравнению с GPL может быть значительным преимуществом.

# 12. Заключение

В отличие от лицензии GPL, которая предназначена для предотвращения коммерциализации открытого исходного кода в проприетарных целях, лицензия BSD накладывает минимальные ограничения на дальнейшие действия. Это позволяет коду BSD оставаться открытым или быть интегрированным в коммерческие решения, в зависимости от изменяющихся потребностей проекта или компании. Другими словами, лицензия BSD не превращается в юридическую "бомбу замедленного действия" на каком-либо этапе процесса разработки.

Помимо этого, поскольку лицензия BSD не обладает юридической сложностью лицензий GPL или LGPL, она позволяет разработчикам и компаниям тратить время на создание и продвижение качественного кода, вместо того чтобы беспокоиться о возможном нарушении лицензирования.

# Библиографические ссылки


- [1] http://www.gnu.org/licenses/gpl.html
- [2] http://archives.cnn.com/2000/TECH/computing/03/28/cyberpatrol.mirrors/
- [3] Open Source: the Unauthorized White Papers, Donald K. Rosenberg, IDG Books, 2000. Quotes are from page 114, "Effects of the GNU GPL".
- [4] В разделе "What License to Use?" книги http://www.oreilly.com/catalog/opensources/book/brian.html


Этот технический документ представляет собой сжатое изложение оригинальной работы, доступной по адресу http://alumni.cse.ucsc.edu/~brucem/open_source_license.htm