**Відкриті Ліцензії на Програмне Забезпечення**

Ліцензія на програмне забезпечення - це правовий інструмент, який визначає використання і поширення програмного забезпечення, захищеного авторським правом. Як правило, ліцензія на програмне забезпечення дозволяє одержувачеві використовувати одну або кілька копій програми, причому без ліцензії таке використання розглядалося б в рамках закону як порушення авторських прав видавця. По суті, ліцензія виступає гарантією того, що видавець ПЗ, якому належать виключні права на програму, не подасть у суд на того, хто нею користується.

**Основні вільні ліцензії**

**GPLv3 (GNU General Public License Version 3)**

Найвідоміша з вільних ліцензій. Не в тому сенсі, що все добре її розуміють, а в тому, що вона найбільше на слуху. Так як вона називається вільної, багато хто помилково вважає, що код, випущений під GPL, можна використовувати як завгодно, а програми можуть / повинні бути тільки безкоштовними. І те, і інше - неправда. GNU GPL - копілефтная ліцензія, і вимагає, щоб вихідні коди похідних робіт були відкритими під нею ж. Тобто, якщо ви вирішите використовувати бібліотеку під GPL в вашому проекті, вам доведеться викласти вихідні вашого проекту під GPL, що зазвичай є неприйнятним для комерційних розробок вам доведеться безкоштовно надавати вихідні проекту кінцевим одержувачам, навіть якщо ви поширюєте продукт за гроші. Так, продаж програми ліцензією цілком дозволена. Надавати початкові можна як разом з програмою, так і окремо. У другому випадку бінарна версія програми повинна містити чіткі інструкції щодо отримання вихідних кодів. Більш докладні пояснення - переведення частини офіціальногоGPL FAQ.

**GPLv2 (GNU General Public License Version 2)**

GPLv3 помітно суворіше і може створити деякі проблеми автору. Наприклад, одна з вимог полягає в тому, що повинна бути надана інструкція по установці зміненого додатка на пристрій. Для додатків під iOS або WindowsPhone, де немає штатної можливості встановити пакет не з магазину, виконати таку вимогу проблематично.

Крім того, варто зауважити, що більшість програм, випущених, під GNU GPLv2, дозволяють використання на умовах більш пізньої версії ліцензії.

**LGPLv3 (GNU Lesser General Public License Version 3, в дівоцтві GNU Library General Public License)**

LGPL - це надбудова над GPL, і вимагає наявності текстів обох ліцензій в проекті. Відмінність LGPL від звичайної GPL полягає в тому, що бібліотеки під цією ліцензією дозволяється використовувати для створення програм під іншими ліцензіями шляхом компонування. Згідно GNU FAQ, динамічна компоновка не вимагає відкривати вихідні коди, а при статичної необхідно надавати вашу програму в об'єктної формі (вихідні коди не обов'язкові), для того щоб користувач міг сам змінити бібліотеку і облінкований бінарник. Втім, як зауважив Athari, це не було обгрунтовано в суді. GNU рекомендує застосовувати цю ліцензію тільки в тому випадку, якщо застосування замість неї звичайною GPL призведе до того, що бібліотеку перестануть використовувати, замінивши на аналогічну.

**GNU AGPLv3 (GNU Affero, GNU Affero General Public License Version 3)**

Її умови фактично складаються з умов GPLv3 з додатковим параграфом в розділі 13, який дозволяє користувачам, які взаємодіють з ліцензованої програмою по мережі, отримувати початковий текст цієї програми. Ми рекомендуємо розробникам подумати про застосування GNU AGPL для будь-яких програм, які зазвичай виконуються в мережі.

Є певні нюанси сумісності з іншими версіями GPL:

Зверніть увагу, що GNU AGPL не сумісна з GPLv2. Вона також формально не сумісна з GPLv3 у вузькому сенсі: ви не можете взяти вихідні тексти, випущені на умовах GNU AGPL, і передавати або змінювати їх, як вам завгодно, на умовах GPLv3, і навпаки. Однак вам дозволено комбінувати роздільні модулі або файли вихідного тексту, випущені під обома цими ліцензіями, в єдиному проекті, що надасть багатьом програмістам дозвіл на всі дії, потрібні їм для того, щоб робити якісь їм завгодно програми.

**MPL v2.0 (Mozilla Public License Version 2.0)**

Для проекту open source варто ще розглянути MPL 2.0. Своєрідна ліцензія, щось середнє між LGPL і BSD. Від LGPL відрізняється відсутністю проблем зі статичним зв'язуванням. Це може виявитися важливим для програм на ЯП, в яких динамічне зв'язування не передбачено.

У разі використання незміненій бібліотеки під MPL 2.0, як частини більшого проекту, потрібно всього лише вказувати, де можна отримати вихідні цієї бібліотеки. Але якщо ви все ж змінюєте код, то зобов'язані надати доступ до зміненого вами коду під все тієї ж MPL 2.0. Тобто, ліцензія копілефтная. Тут невелике уточнення від Athari:

Ліцензією MPL заражаються файли, а не проекти, на відміну від (L) GPL. Якщо змінити файл, він повинен залишитися під MPL. Якщо додати - обмежень немає.

У разі, якщо проект під GNU GPL, то необхідно зробити використовуваний в ньому код під MPL 2.0 доступним відразу під обома ліцензіями.

**EPL-1.0 (Eclipse Public License Version 1.0)**

На прохання kidar2 додаю ліцензію EPL. Це копілефтная ліцензія, але вона не сумісна з GNU GPL.

При поширенні в формі вихідного коду програма повинна бути доступна під ліцензією EPL.

Автору дозволяється поширювати програму в формі об'єктного коду під власною ліцензією, за умови, що: ця ліцензія діє згідно з умовами EPL, явно відмовляється від будь-яких гарантій і відповідальності від імені всіх авторів, вказує, що вихідні коди програми доступні у цього автора і пояснює, як їх отримати.

Застосування до свого проекту: копія ліцензії повинна бути включена в усі копії програми.

**Ms-PL (Microsoft Public License)**

Про ліцензію нагадав sensboston. Копілефтная ліцензія, несумісна з GPL. За змістом схожа з EPL, але написана набагато, набагато більше человеческм мовою. Найкоротша з присутніх в цій статті копілефтних ліцензій.

Володіє навіть слабшим копілефту, ніж EPL: якщо ви поширюєте вихідні коди проекту, що містять код під Ms-PL, то всі вихідні коди проекту повинні поширюватися під Ms-PL. При цьому, поширення в формі об'єктного коду або бінарної формі дозволяється під будь-який ліцензією, не порушує Ms-PL. Крім того, ви зобов'язані зберігати всі авторські права, патенти, торгові марки та вказівки авторства оригінального коду. Так, ліцензія регулює патентні відносини.

Для застосування до свого проекту: скопіюйте текст ліцензії в ваш проект (наприклад, в файл LICENSE) і поширюйте його разом з ним.

**MIT**

Існує міф, що ліцензія MIT існує. Справа в тому, що MIT (Massachusetts Institute of Technology) використовував багато різних ліцензій. Той текст, який зараз називають ліцензією MIT, в оригіналі був ліцензією Expat, а ще раніше становив більшу частину ліцензії X11. Ця ліцензія - дозвільний, без копілефту. Вона дозволяє використати або відредагувати коду практично будь-яким чином, за умови, що текст самої ліцензії та зазначення авторства нікуди не зникнуть, навіть якщо ви розіб'єте початковий проект на частини. Також незаперечне достоїнство цієї ліцензії - невеликий розмір. Як недолік відзначають відсутність регулювання патентних відносин. Через це замість неї GNU рекомендують використовувати іншу дозвільну ліцензію - Apache 2.0, а MIT пропонують використовувати лише для невеликих проектів. Проте, з дозвільних ліцензій ця, мабуть, найвідоміша.

Для її застосування до свого проекту створіть текстовий файл LICENSE і помістіть текст ліцензії туди, а також не забудьте замінити дані в рядку з копірайтом на вірні. Багато додатково вказують повний текст ліцензії в шапці кожного файлу вихідного коду.

**Apache 2.0**

Найбільш сучасна і збалансована з дозвільних ліцензій. Написана людською мовою, але з оглядкою на сучасне правозастосування, зокрема, згадані вище патентні відносини (пункт 3 ліцензії). GNU радять застосовувати саме цю ліцензію, коли вам необхідна дозвільна ліцензія.

Для застосування ліцензії Apache 2.0 до вашого проекту, потрібно додати в нього файл LICENSE, що містить текст ліцензії. Крім того, в APPENDIX ліцензії нам пропонують додавати в якості шапки в кожен файл вихідного коду наступний текст:

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions

**Суспільне надбання (Public Domain)**

Це, звичайно ж, не ліцензія. Але багато хто розглядає переклад твору в суспільне надбання як спосіб скласти з себе майнові права. Періодично люди намагаються зробити це і в відношенні програм. Зазвичай просто пишуть, що проект знаходиться в Public Domain і радіють, що всі зможуть ним користуватися. Але насправді це не так! У різних країнах суспільне надбання різний. Причому, в більшості з них закон явно не передбачає механізмів для перекладу твору в суспільне надбання за бажанням автора! Наприклад, в Росії перехід до суспільного надбання визначено тільки по закінченню терміну дії захисту авторського права. У США, наскільки мені відомо, дострокової процедури теж не передбачено. Крім того, можуть бути складності і з використанням творів, і є суспільним надбанням. Скажімо, в Уругваї, для того, щоб використовувати твір з суспільного надбання, вам знадобиться сплатити мито державі і отримати дозвіл спеціальної комісії, яка упевниться, що ви не будете на використання цієї роботи якось непотребно. Внаслідок подібних нюансів, з'явилися ліцензії подібні наступної.

**CC0 (Creative Commons CC0)**

Creative Commons CC0 - ліцензія, яка намагається перевести проект в суспільне надбання в максимальній формі, дозволеній законом. А якщо закон не дозволяє це зробити, автоматично застосовує положення дозвільної ліцензії. GNU рекомендує застосовувати CC0 в тому випадку, якщо ви хочете перевести вашу роботу в суспільне надбання.

**Unlicense**

Про ліцензію нагадав Athari. Ця ліцензія з'явилася шляхом копіпаст тексту про передачу в суспільне надбання і відмови від прав (waiver) проекту SQLite і відмови від гарантій з ліцензії MIT. Аналогічно ліцензії CC0, Unlicense намагається перевести роботу в суспільне надбання і послужити у вигляді ліцензійного договору на випадок, якщо цього не відбулося. Однак, ця ліцензія менш опрацьована, ніж CC0, через що може бути нелегальної. Ось в цьому питанні на stackexchange докладніше. Якщо коротко, там зазначено, що ліцензія явно нелегальна, наприклад, в Німеччині, так як там, схоже, немає поняття суспільного надбання. А Unlicense, на відміну від CC0, не відмовляється від перекладу в суспільне надбання для випадку, коли це суперечить закону. Крім того ліцензія як мінімум нелогічна (або навіть суперечлива), так як передача в суспільне надбання, заявлена ​​в першому рядку, в разі успіху робить невалидность параграфи, які йдуть за нею.

Для застосування Unlicense потрібно додати файл з текстом ліцензії до вашого проекту. Автори ліцензії рекомендують назвати файл UNLICENSE.

**Copyright у ісходних файлах**

Напевно, ви помітили, що багато ліцензії пропонують розміщувати певний текст у вигляді коментаря в шапці файлу? Якщо це є обов'язковою вимогою, то тоді йому потрібно слідувати. Але наскільки необхідний подібний текст, якщо явного вимоги ліцензія не пред'являє?

Хороші новини: в такому випадку ліцензію і навіть копірайт зовсім не обов'язково вказувати в шапці файлу. Ваша робота і так ваша, для підтвердження цього вказувати копірайт немає необхідності. Підтверджувати авторство або володіння правами вам все одно доведеться іншими способами, а текст ліцензії може перебувати в окремому файлі.

Але все ж такий заголовок краще мати. Основні причини такі:

Він чітко показує, що права на код комусь належать. Відмазки типу «я не знав, там не написано, не зміг знайти, не помітив» вже не покатають. Інакше кажучи, наявність такого заголовка запобігає випадкове неправомірне використання коду, а також може збільшити відповідальність за навмисне.

Дає можливість ідентифікувати власника прав на код, щоб зв'язатися з ним в тому числі і з питань правомірності використання цього коду.

Між іншим, це має сенс не тільки для open-source проекту. Вказівка ​​власника прав і авторства може допомогти і пропрієтарного коду в тому випадку, якщо він будь-яким чином витече в мережу.

Якщо ви вирішили, що повідомлення про права на файл вихідного коду вам необхідно, то ось що воно повинно містити в ідеалі:

Copyright - так як в деяких країнах одного символу копірайту недостатньо для юридичної значимості повідомлення.

© - символ копірайту, в більшості країн він необхідний і достатній для надання юридичної значимості повідомленню. Простий букви c в дужках (c) для цього може бути недостатньо. Використовуйте вже Unicode!

2007, 2009, 2010 - 2012 - «роки життя» коду, перше число - рік, коли продукт був вперше опублікований, далі - роки, коли код оновлювався. Якщо роки, коли код у файлі правили, йдуть не підряд, то їх потрібно вказувати через кому.

John Doe - ім'я власника авторських прав. Чи не автора, це можуть бути різні особи! Може бути ім'ям людини, назвою компанії, або іменами декількох чоловік. Правда, в останньому випадку краще зробити окремий рядок на кожну людину.

All rights reserved - «все права захищені», означає, що зазначені особи мають усі права на код. Додаткове посилення повідомлення копірайту, в разі володіння винятковими правами.

Якщо можливо, то дати посилання на ліцензійний договір або вказівку, де його шукати.

Вказати контактні дані.

Зверніть увагу, що ім'я автора в повідомлення не входить. Автора / авторів можна вказати окремо, наприклад, на наступному рядку, у вільній формі (наприклад Author: Jane Doe).