

A szabad szoftverek testreszabása

Készítette a Közigazgatási és Igazságügyi minisztérium E-közigazgatási
Szabad Szoftver Kompetencia Központja
Budapest, 2013



A projekt az Európai Unió támogatásával, az Európai
Regionális Fejlesztési Alap társfinanszírozásával valósul meg.

Kódszám: EKOP–1.2.15

Ez a Mű a Creative Commons Nevezd meg! – Így add tovább! 3.0 Unported
Licenc feltételeinek megfelelően szabadon felhasználható.

A dokumentum legfrissebb változata letölthető a honlapunkról:

<http://szabadszoftver.kormany.hu/>

Tartalomjegyzék

Szabad szoftverek.....	2
Szabad szoftverek módosíthatósága.....	2
Szabad szoftveres licencek.....	4
Néhány elterjedtebb szabad szoftveres licenc.....	4
BSDL – Berkeley Software Distribution Licence.....	4
GPLv2 – GNU General Public Licence v2.....	4
GPLv3 – GNU General Public Licence v3.....	5
LGPL – GNU Lesser General Public Licence.....	5
Hasonlóságok.....	5
Különbségek.....	6
Egy példa.....	7
Összefoglalás.....	8

A szabad szoftverek testreszabása

Szabad szoftverek

Jelen tanulmányban a nyílt forráskódú, szabadon felhasználható és terjeszthető szoftverekről lesz szó. Az egyszerűség kedvéért és a szóismétlések elkerülése érdekében ezekre hol nyílt forráskódú, hol szabad szoftverként fogunk hivatkozni – a kifejezéseket egymás szinonimáiként használva. (Időnként szükséges lesz a „másik oldalról” beszélni, ezeket szokták tulajdonosi szoftvereknek is nevezni, itt az egyszerű nem-szabad szoftver kifejezést használjuk.)

Kezdjük egy rövid fogalmi magyarázattal. Nagyon leegyszerűsítve a dolgokat, szabad szoftverekről akkor beszélünk, ha a szoftver fejlesztője, terjesztője nem köti meg a felhasználó kezét. Sem a használhatóságot, sem a szoftver módosíthatóságát, sem a továbbterjesztést illetően. (Elsősorban a továbbterjesztést illetően jelentős különbségek találhatók az egyes szabad licencek között.)

Egy szoftver felhasználhatósági korlátja alatt olyan megkötésekre gondolunk, amelyekkel nap mint nap lehet találkozni, mint például:

- **időbeli korlát:** az alkalmazás maximum egy hónapig használható, ezt követően regisztrálnunk kell, vagy meg kell vásárolni a használat jogát; esetleg az adott program 10-szeri elindítást engedélyez.
- **funkcionalitásbeli megszorítás:** például csak egy adott méretkorlát (pl. 1000 adatrekord) eléréséig használható az adatbázis-kezelő; vagy csak 1 megapixeles képek feldolgozására használható a képmánipuláló eszköz.

Előfordulhatnak a használatot nehezítő vagy a végeredmény szélesebb körű használatát nehezkessé tevő funkciók is:

- a fájlkezelő alkalmazás indulásakor a felhasználó nyomja le a véletlenszerűen kiválasztott billentyűk egyikét, vagy kattintson egy rendszeresen máshol felbukkanó gombra
- a képfeldolgozó szoftver minden képre jól látható vízjelet tesz, a bemutatókészítő program pedig az elkészült prezentációba véletlenszerűen felbukkanó „Készült XYZ szoftver korlátozottan használható verziójával” diát illeszt be

Nyílt forráskódú szoftver esetén nincsenek meg ezek a korlátok. Nincsenek, hiszen ha egyszer a forrás elérhető, akkor kellő tudás birtokában bárki kiszedhetné ezeket a kényelmetlenséget okozó, bosszantó funkciókat.

Szabad szoftver esetén a módosíthatóságon kívül a módosított szoftver szabad terjeszthetősége is fontos további szempont. Azaz nem csak az engedélyezett, hogy az, aki az adott szoftver tudásával vagy működésével nem elégedett, saját magának átalakítsa, hanem a licenc arra is lehetőséget biztosít, hogy ha valaki aktívan tenni is szeretne a kérdéses szoftver jobbá tétele érdekében, azt megtehesse, és ezeket a javításokat terjeszthesse.

Szabad szoftver esetén – mivel a forráskód elérhető és módosítható – elvi (és gyakorlati) szinten is van lehetőség az adott alkalmazást módosítani oly módon, ahogyan az az adott esetben szükséges, majd a módosított eredményt terjeszteni (nem-szabad szoftverek esetén még ez az elvi lehetőség sincs meg).

Szabad szoftverek módosíthatósága

Tetszőleges szoftver eleve meglevő tudásának kihasználhatósága legfeljebb a felhasználó szorgalmán múlik. (Az effektív, hatékony használathoz nyilvánvalóan azt meg kell tanulni – itt játszik szerepet a szorgalom.) Ha valakinek szüksége van egy olyan szolgáltatásra, amivel a kérdéses szoftver rendelkezik, akkor máris használatba veheti, és (jobb esetben) kötöttségek nélkül, megelégedetten használhatja is azt.

Míg a felhasználók jelentős része a számítástechnikában kevésbé járatos, addig sokan komolyabb informatikai végzettséggel – vagy ami fontosabb, tudással – rendelkeznek. Ezen felhasználók számára – ahogy korábban már utaltunk rá – a szabad szoftverek második aspektusa, a forrás módosíthatósága is fontos lehet. Mivel nem születik senki programozónak, és később is csak kevesen válnak azzá, ez a lehetőség nyilván egyéni, otthoni felhasználók esetén csekély jelentőségű. Természetesen nem csak egyéni felhasználók léteznek. Kisebb-nagyobb vállalatok, állami hivatalok esetében azonban már érdemes elgondolkodni a szabad szoftverek módosíthatóságáról, mint pozitívumról. Miért fontos a módosíthatóság? A válasz egyszerre könnyű és nehéz:

- Könnyű, mert mindenki, aki már használt számítógépet, könnyedén tudna példát találni olyan esetre, amikor egy általa használt szoftver esetén a szoftver korlátaiba ütközött. Ilyen helyzetben nem-szabad szoftver esetén lehetőségeink szűkösek: vagy együtt élünk a problémával, vagy választunk másik szoftvert, amely az adott tudással rendelkezik – és máris lehet újratulni a használatot –, vagy pedig jelezzük a szoftver fejlesztőjének igényünket, és utána türelmesen várakozunk, hogy a szoftver egy valamikor később megjelenő verziójában – rosszabb esetben esetleg pénzért, vagy soha – elérhető legyen a hón áhított szolgáltatás. Ellenben szabad szoftver esetén a fentiekén kívül van még legalább egy lehetőségünk. Vagy mi magunk elvégezzük, vagy valaki más, szoftverfejlesztői tudással megáldott emberrel – vagy akár egy egész csapattal – elvégeztetjük a szükséges módosításokat.
- Más oldalról nézve nehéz a válasz, hiszen az emberek többsége sem nem szoftverfejlesztő, sem nem ismer olyat, aki az lenne; ily módon számára nem érzékelhető a fentebb előnyként feltüntetett lehetőség. Azaz ez csak egy semmilyen pozitívumot nem hordozó tulajdonsága a szabad szoftvereknek, csak látszólagos előny a végfelhasználók számára. Ellenben egy cég, egy állami szervezet esetén ez nem lehet probléma. Ahogyan az adott cég kiválasztotta az addig használt számítógépes környezetet és a rendszer üzemeltetőjét, ugyanolyan módon kiválaszthat egy fejlesztőt vagy fejlesztőgárdát is, aki/amely a szükséges módosításokat elvégzi számára.

Egy pillanatra vizsgáljunk meg egy eddig figyelembe nem vett szempontot. Alapigazság, tökéletes szoftver nincs. Ugyanígy olyan sincs, amelyik minden – az adott feladatkörben reálisan elvárha-

tó – feladatot tökéletesen ellátna. De ha esetleg valaki megtalálná a világon az egyetlen tökéletes szoftvert, még mindig van egy pont: ha más nem is, de a kezelőfelület az esetek jelentős részében egyik vagy másik felhasználónak nem fog kézre esni (illetve mivel mindenki saját eljárásokat szeret kidolgozni egy feladat elvégzéséhez, máris oda jutottunk, hogy az egyéni testre szabhatóság és a program módosíthatósága jó dolog – főleg, ha van idő megtanulni, illetve mindenkinek magának beállítani a szoftvert). A felhasználók szoftverekkel kapcsolatos problémái valószínűleg legtöbbször a programok hibás működéséből, ritkábban a hiányzó funkciókból, esetleg ergonómiai okokból erednek.

Ha egy szoftver nem tud mindent, ami a feladat elvégzéséhez szükséges, akkor valószínűleg van olyan funkció, amelyikhez egy másik szoftverre van szükségünk – és ebben az esetben elérkeztünk egy újabb problémához: feladatunk egyik részét elvégezzük az egyik szoftverrel, majd a feladat másik részét a másik szoftverrel. Ehhez azonban szükségünk van arra, hogy az adatok a két szoftverrendszer között oda-vissza mozgathatóak legyenek. Vannak olyan helyzetek, amikor ez látszólag nem probléma, például a legtöbb képfeldolgozó képes TIFF formátumú állományok kezelésére, így megtehetjük, hogy ebben a formátumban mentjük a módosítás után a fotót az egyik programban, majd betöltjük a másodikba, és így tovább, de ez így meglehetősen kényelmetlen, igaz a probléma elvileg megoldható. Más esetekben azonban nem ennyire egyszerű a helyzet, és lehet, hogy hosszútávon mindenki jobban jár, ha inkább kibővít(tet)jük az eredetileg kiválasztott alkalmazást (vagy egy másik szoftvert választunk). Ez a téma viszont önmagában is elég terjedelmes, így ezt egy másik, a „Szabad szoftverek, nyílt formátumok és a nyílt szabványok”¹ című tanulmányban tárgyaljuk.

Amennyiben valaki elfogadja, hogy egy szoftver használata esetén az adott szoftver módosíthatósága számára előnyökkel jár, meg kell vizsgálnia a lehetőségeket. Milyen szoftverek szükségesek számára, az elérhetőek közül melyek megfelelőek? Ha nincs megfelelő, akkor melyik az (vagy melyek azok), amely valószínűleg legkönnyebben alakítható át a számára szükséges tudására. A vizsgálat során a szoftver későbbi felhasználásától függően az egyik vizsgálandó feltétel kell legyen az adott szoftver által használt licenc.

Szabad szoftveres licenck

A továbbiakban röviden ismertetjük néhány ismertebb szabad szoftveres licenc tartalmát, majd pár szóban összevetjük őket. Azt azonban fontos megjegyezni, hogy az itt szereplő megfogalmazások jogi szakszövegeknek nem hivatalos fordításaiból készült, „kivonatossnak” is csak jóindulattal nevezhető magyarázatok, ezen kívül nem vizsgáltuk az egyes licenceknek a Magyarországon érvényben levő törvényekkel való harmonizációját. Azaz minden esetben vagy az eredeti licencet, vagy annak hivatalos fordítását javasolt elolvasni és értelmezni.

Néhány elterjedtebb szabad szoftveres licenc

A szoftverfejlesztők is emberek, így az évek során több különböző, egymásra kisebb-nagyobb mértékben hasonlító szabad szoftveres licenc alakult ki. Ezen licencek mindegyikének ismertetésétől hely- és időhiány miatt eltekintենk, de néhány fontosabbra azért sort kerítünk. Az egyes licencek részletesebb kifejtése egy másik tanulmányban kap helyet (lásd „A fontosabb szabad szoftver licencek jellemzői”² című tanulmányt). Fontos ezen kívül megjegyezni, hogy például az FSF (Free Software Foundation)³ csak az angol nyelvű eredeti verziót tekinti hivatalosnak, semmilyen egyéb nyelvű fordítást nem.

¹<http://szabadszoftver.kormany.hu/sajat-tanulmanyok#szsz-nyilt-formatum-szabvany>

²<http://szabadszoftver.kormany.hu/sajat-tanulmanyok/#szsz-licenck-jellemzoi>

³<http://www.fsf.org/>

BSD – Berkeley Software Distribution Licence

A kaliforniai Berkeley Egyetemen (UCB – University of California, Berkeley) ⁴ kifejlesztett BSD UNIX rendszerekkel elterjedt licenct utoljára 1999-ben módosították (1989, más források szerint 1988 az eredeti megjelenés dátuma⁵). Gyakorlatilag az ilyen licenct használó szoftverek esetén egyetlen megkötés van: nem állíthatjuk azt, hogy a szoftver a mi termékünk – terjesztés esetén jelezniük kell, hogy mások munkájára alapoztunk, az eredeti fejlesztő jogi nyilatkozatát is mellékelniük kell. Engedélyezi a szoftver módosítását, és akár nyílt, akár zárt licenct alatt történő további terjesztését. Ez természetesen azt jelenti, hogy egy BSD-t használó szoftver fejlesztése esetén egyáltalán nem biztos, hogy a későbbi, mások által elvégzett módosítások kijutnak a „világba”, azaz könnyedén lehet segítségével zárt, tulajdonosi szoftvert előállítani. (Egyes magyarázatok szerint talán ez a leginkább megengedő, a legnyíltabb nyílt forrású licenct.) Valószínűleg a legismertebb BSD-t használó szoftver a FreeBSD⁶ néven ismert operációs rendszer.

GPLv2 – GNU General Public Licence v2.

A Free Software Foundation által 1989-ben megalkotott licenct (GPLv1⁷) 1991-ben létrehozott módosítása, melynek – minket érintő – legfontosabb kitétele, hogy noha akár bináris formátumban is terjeszthetjük az ilyen licenct használó szoftvereket, sőt pénzt is kérhetünk értük, ettől függetlenül a későbbi módosításoknak szintén GPL alatt elérhetőeknek kell lenniük, ily módon biztosítva azt, hogy ezek a módosítások szintén elérhetőek és szabadon felhasználhatóak. Azaz aki GPL-es kódot használ programjában, az köteles a saját programját is GPL alatt elérhetővé tenni, és a forráskódot (legrosszabb esetben csak kérésre ⁸) elérhetővé tenni. A gyakorlatban az ilyen szoftver forráskódját vagy szabadon elérhetővé teszik az interneten (és általában megadják az elérhetőséget a dokumentációban), vagy pedig a bináris terjesztés részeként a CD/DVD telepítőn mellékelik. Legismertebb és leghatárterjedtebb GPLv2-t használó rendszer a Linux kernel⁹.

GPLv3 – GNU General Public Licence v3.

A fenti GPLv2 2007-es módosítása, amely egyebek mellett tiltja (a korábban nem szabályozott) kétoldalú kereskedelmi megállapodásokat a GPLv3-at használó szoftverekkel kapcsolatban. (Ilyen megállapodást kötött többek között 2006-ban a Microsoft és a Novell¹⁰, illetve 2007-ben a Microsoft és a Xandros¹¹.) Szintén a módosítások közé tartoznak a szoftverszabálmakkal kapcsolatos kitételek, illetve az egyes cégek által alkalmazott, hardver korlátozásokat alkalmazó¹² trükkök kikerülésére vonatkozó részek. Példa GPLv3-t használó, elterjedt szoftverre a GNU Compiler Collection szoftvercsomag minden, a 4.2.1 verzió utáni kiadása – a korábbiak GPLv2-t használnak.¹³

LGPL – GNU Lesser General Public Licence

Kereskedelmi szoftvereket gyártók számára a GPL egyik komoly korlátozó tulajdonsága, hogy GPL-es kódot felhasználva a végleges szoftvernek is GPL-esnek – azaz nyílt forráskódúnak - kell lennie (ily módon a zseniális algoritmusok, trükkös megoldások a konkurencia számára is elérhető-

⁴ <http://berkeley.edu/>

⁵ http://en.wikipedia.org/wiki/BSD_licenses

⁶ <http://www.freebsd.org/>

⁷ http://en.wikipedia.org/wiki/GNU_General_Public_License

⁸ Az elérhetőség mindenkinek mást jelent. Jelen tanulmány egyik szerzője nem tartja jó megoldásnak azt, hogy a módosítások eléréséhez pl. elektronikus levélben kell fordulni a céghez, és csak azután lehet hozzáférni (például azért mert ha a cég jogutód nélkül megszűnik, akkor reálisan elveszik a módosítás. Ráadásul sokak számára visszatetsző, hogy „könyörögniük” kell valamiért, ami a jog szerint nekik „jár”).

⁹ <http://www.linux.org/>

¹⁰ http://en.wikipedia.org/wiki/Microsoft-Novell_agreement#Agreement_with_Microsoft

¹¹ <http://www.xandros.com/xandrosandmicrosoft.html>

¹² <http://en.wikipedia.org/wiki/Tivoization>

¹³ <http://gcc.gnu.org/ml/gcc-announce/2007/msg00003.html>

vé válnak). Ez különböző eljárásgyűjtemények („függvénykönyvtárak”) használata esetén nagyon szűk teret enged a szoftverfejlesztőknek, ezért jött létre az ezt a kötöttséget nem tartalmazó LGPL. Bizonyos technikai feltételek teljesülése esetén (statikus szerkesztés helyett úgynevezett. dinamikus szerkesztést használva – .DLL-ek, JAR-fájlok, .so osztott függvénykönyvtárak használata) kereskedelmi, zárt forráskódú szoftverek is használhatnak LGPL-es komponenseket. LGPL-es szoftverre példa a legtöbb Linux terjesztésben használatos C-eljárásgyűjtemény, a GLIBC; vagy a meglehetősen sok grafikus felületű alkalmazás fejlesztéséhez használatos grafikus függvénykönyvtár, a GIMP Toolkit – más néven GTK+¹⁴.

Hasonlóságok

Fenti licencek esetében legfontosabb hasonlóság a kód módosíthatósága. Mindegyik lehetővé teszi, hogy a saját magunk számára szükséges módosításokat elvégezzük. Azaz bármely fenti licenc használata esetén egy programozni tudó magánszemély minden további nélkül kijavíthatja a számára bosszantó programhibát, illetve hozzáfeshozhatja a neki hiányzó szolgáltatást. De mi a helyzet akkor, ha az említett programozó szomszédja – aki nem programozó –, szintén szeretné használni a javított verziót. Odaadható-e a módosított program? Kell-e mellékelni a forráskódot? Mindez magánszemélyek esetében jelenleg nem különösebben problémás, valószínűleg még senkit nem pereltek be – pláne nem ítélték el – szabad szoftveres licenc megsértésért. Ellenben mindez sokkal fontosabbá válik akkor, amikor nem egy magánszemély, hanem egy cég, vagy állami szerv döntéséről van szó.

Különbségek

Mint látható, komoly különbségek vannak az egyes licencek között a módosíthatóságot, illetve a továbbfejleszthetőséget illetően. Van olyan, amelynél azt csinálunk, ami jólesik. Van ahol magát a módosítatlan eredeti forrást kell odaadnunk, és odaadhatjuk hozzá a mi javításainkat – aztán mindenki hozzá belőle létre a módosított verziót; és így tovább.

Mit jelentenek mindezek a különbségek a gyakorlatban?

Egy szoftverfejlesztő cég számára, amelynek kőkemény konkurenciával kell számolnia, tetszőleges szoftverfejlesztés esetén mások BSDL alatt elérhetővé tett programjai kincsesbányaként működnek. Készen van, (jó esetben) használható, azt csinál vele, amit akar, a maximum, hogy a végtermék leírásában szerepeltetni kell azt a szöveget is, hogy „Copyright © ÉÉÉÉ.HH.NN, X.Y”.

Ugyanezen szoftverfejlesztőnek amennyiben ő alkalmazza a BSDL-t, számíthatnia kell arra, hogy később esetleg jön valaki más, és az ő termékét – esetlegesen csak minimális változtatással – zárt forráskódú, tulajdonosi szoftverként pénzért árusítja, ez által rontva a cég nyereségességét, rosszabb esetben esetleg ügyesebb marketinggel, jobb reklámokat alkalmazva kiszorítja az eredeti terméket a piacról és akár csődbe is juttathatja az eredeti fejlesztőket. Előfordulhat, hogy valamely, a programban alkalmazott kommunikációs vagy adattárolási mechanizmust kicsit módosítva egy olyan módosított verzió jön létre, amelyik az eredetivel nem kompatibilis. Amennyiben ezt az új verziót sikerül elterjeszteni, akkor a szoftver eredeti fejlesztőjének a későbbiekben alkalmazkodnia kell ehhez a módosított formához – esetleg akkor is, ha a módosításoknak valójában technikai okai nem voltak, esetleg az eredeti elképzelésekkel homlokegyenest ellentétes elvek alapján álltak elő.

Amennyiben egy szoftverfejlesztő eredetileg zseniális ötletével a világ megváltására törekvése közben elfeledkezik a pénzügyi realitásokról, és a világ jobbá tétele érdekében BSDL-t választ szoftveréhez, akkor nagy eséllyel elbukik, hisz bárki bármit tehet termékével. Természetszerűleg abban a pillanatban, ha a pénzügyi realitások nem érdeklik – mert nem ebből él, és/vagy egyáltalán

¹⁴ <http://www.gtk.org/>

nem érdekli, hogy a későbbiekben ki mit csinál fejlesztésével, akkor nyugodtan megteheti, hogy BSDL alatt terjeszti kódját.

Mivel az üzleti világban meglehetősen elterjedt gyakorlat, hogy egy tőkeerős vállalat pénzügyi erejével (vissza)élve nyomja el ellenfeleit (egy cég felvásárlása, majd termékének elsorvasztása meglehetősen gyakran alkalmazott technológia), profitorientált környezetben a kereskedelmi termékhez a BSDL használata nem jó választás.

Azaz BSDL választása akkor javasolható, ha az eredeti fejlesztő számára sem lelki, sem anyagi törést nem okoz a fentebb felvázolt lehetőségek bekövetkezése, esetleg bízik abban, hogy a későbbi módosításokat elkövetőknek nem érdekük inkompatibilis változtatásokat eszközölni.

Amennyiben fent említett szoftverfejlesztő (vagy aki a fejlesztőt megbízza, és ezen igényét a szerződésben is rögzíti) GPL-t választ, akkor elvben sokkal hasznosabb a közösség számára. Nem megengedett az adott kódot bezárni, és nem-szabad szoftver fejlesztéséhez felhasználni – hiszen muszáj a módosításokat elérhetővé tenni. Ha valakinek szüksége van valamilyen addig nem létező szolgáltatásra, és azt kifejleszti vagy kifejleszteti, akkor ezt a fejlesztést a világon mindenki használhatja, hisz a módosítások forrását át kell adni minden felhasználónak, akik ezt szabadon nyilvánosságra hozhatják. Meglehetősen gyakori dolog, hogy egy ígéretes fejlesztéshez a kezdeti stádium után mások is csatlakoznak, mindenki hozzáteszi a termékhez a számára szükséges részeket, kijavítja az őt valamely feladat elvégzésében akadályozó hibákat. Természetesen a fejlesztést koordinálni is kell valakinek, ez általában maga az eredeti fejlesztő, de az esetek jelentős részében a későbbiek során váltanak – amikor az eredeti fejlesztő elveszti érdeklődését, esetleg egyéb elfoglaltságai miatt már nem tud annyi időt tölteni a programmal, amennyit vezetőként kellene.

Amennyiben valaki mégis megpróbál GPL-t használó szoftvert zárt termékben alkalmazni, akkor számítani kell arra, hogy amennyiben ez kiderül, annak általában komoly sajtóvisszhangja lesz – amely nem kifejezetten segíti a cég jó hírnevének megőrzését. Rosszabb esetben a presztízsveszteségen kívül jogi lépésekre is számíthat.

Egy szoftver szabad módosíthatóságának természetesen lehet negatív következménye is. Könnyen előfordulhat, hogy két vagy több cég, szervezet választja egy adott feladathoz ugyanazt az alkalmazást, majd pedig a számukra hiányzó funkcionalitást – mivel nem tudnak egymásról – szépen sorban, mindegyikük kifejleszti vagy kifejleszteti magának. Ezt a problémát nem reménytelen kiküldeni, de mindenképpen odafigyelést igényel.

Az egyik megoldás, hogy mielőtt valaki/valami módosításokat eszközölné, megpróbálja felvenni a kapcsolatot az eredeti szoftverfejlesztővel – legyen az egyetlen ember, vagy egy közösség, esetleg egy sok száz főt foglalkoztató cég –, és megpróbálja kipuhatolni, hogy létezik-e már valaki, aki az adott funkcionalitás fejlesztésébe kezdett, és amennyiben igen, a folytatásban igyekszik vele együttműködni. Amennyiben ilyen módosítás még nem elérhető, akkor jelezni javasolt, hogy mi az, ami készülöben van, majd pedig a módosítást annak elkészülte esetén érdemes eljuttatni az eredeti fejlesztőkhöz. Ez nem minden esetben könnyű (vagy egyáltalán lehetséges), de még ha ez nem is sikerül, abban az esetben is javasolt azonban a módosításokról szakmai fórumokon értesíteni az érdeklődőket. Természetesen javasolt a módosításokat valamilyen formában elérhetővé tenni.

Azt mindenképpen újra és újra hangoztatnunk kell, hogy ez a lépés – a módosítások elérhetővé tétele –, bizonyos esetben nem opcionális, hanem kötelező lépés. Mint korábban már utaltunk rá, míg például a BSDL erre vonatkozóan semmilyen kötelezettséget nem ír elő, addig az ilyen szempontból sokkal szigorúbb GPL kötelezően előírja a módosított forráskódnak, illetve maguknak a módosításoknak az elérhetővé tételét, és a Free Software Foundation külön erre a célra fenntartott e-mail címen¹⁵ várja az ezen kötelezettség megsértésével kapcsolatos bejelentéseket¹⁶.

¹⁵ [mailto: <license-violation@gnu.org>](mailto:license-violation@gnu.org)

¹⁶ <http://www.gnu.org/licenses/gpl-violation.html>

Az igazsághoz tartozik, hogy nem egy esetben előfordult már, hogy kisebb-nagyobb cégek esetében derült ki GPL-sértés, erről többek között pl. a <http://gpl-violations.org> címen lehet tájékozódni.

Egy példa

A korábbiakban, a különböző licencek áttekintése kapcsán elhangzott, hogy pl. a BSDL és a GPL jellegű licencek között az egyik – talán a legényegesebb – különbség, hogy mit tehetünk meg a módosításainkkal. Míg a BSDL megengedi a módosítások bezárását, addig a GPL ezt igen szigorúan szabályozza, és egyértelműen tiltja.

A kaliforniai Berkeley egyetemen kifejlesztett BSD UNIX operációs rendszer BSDL alatt volt elérhető. A forráskódot (amely az akkori idők – 70-es, évek vége, 80-as évek eleje – elterjedtebb számítógépein működőképes volt) némileg kibővítve és módosítva, egy William Jolitz nevű számítógépes szakember – Lynne Jolitz nevű, szintén szakmabeli felesége segítségével – előállított egy olyan verziót, amely az akkortájt egyre elterjedtebb Intel x86 processzort használó PC-ken is működőképes volt (amennyiben a gép legalább Intel 80386-os, vagy annál modernebb processzorra rendelkezett). Az ily módon létrehozott szoftvert aztán 386BSD néven szabadon elérhetővé tette az interneten¹⁷. Mivel a fejlesztőknek nem ez volt a hivatalos munkája, a későbbiekben egyre kevesebb időt tudtak rááldozni, így ezt a munkát felhasználva létrejött több különböző terjesztés. Van ezek között olyan (a korábban már emlegetett FreeBSD), amelyik az eredeti 386BSD célkitűzésének megfelelően elsősorban a „PC”-t tekintette legfontosabb támogatandó hardverplatformnak, míg olyan is akad, amelyik még régebbre nyúlt vissza, és az eredeti BSD hagyományoknak megfelelően inkább a világ minden elérhető hardverplatformját szeretné támogatni (NetBSD¹⁸). Ezek szabad szoftverek maradtak azóta is, és az eredetitől csekély mértékben különböző BSDL alatt érhetőek el a mai napig. Ugyanakkor létrejött egy olyan cég (BSDI¹⁹), amelyik ugyanezt a kódbázist felhasználva, azt természetesen a saját igényeinek megfelelően módosítva, kereskedelmi terméket hozott létre – az eredeti szoftver nevéhez kísértetiesen hasonló BSD/386 néven –, amely terméket aztán elsősorban a vállalati szférában (például számítógépes rendszerek határvédelmi eszközeként, tűzfalnak) pénzért árusította. Figyeljük meg, hogy az adott esetben ezt jogi értelemben nem tekinthetjük a forráskód ellopásának, hisz az eredeti licenc az ilyen jellegű felhasználást minden további nélkül engedélyezi.

Összefoglalás

Nyílt forráskódú, szabad szoftver használata esetén nem szabad egyedüli szempontnak tekinteni, hogy megkapható ingyen, és nem kell licenccéget fizetni valamely cégnek. Hosszú távon sokkal fontosabb az, hogy lehetőség van az alkalmazást pontosan azokkal a funkciókkal felvértezni, amelyekre szükségünk van. Bármely előforduló hiba kijavításának lehetősége adott – hisz a forráskód a kezünkben van. Ennek természetesen ára van – adott esetben meg kell fizetni azt a programozót vagy fejlesztőgárdát, aki a testreszabást, a javítást elvégzi. Ez rövid távon nem biztos hogy olcsóbb, mint a tulajdonosi szoftver használatából adódó költségek, de hosszú távon valószínűleg olcsóbb a mindenkori igényeket kielégítő – és csak a szükséges szolgáltatásokat nyújtó, valamint remélhetőleg egyre javuló minőségű program használata.

Vizsgálatos tudatosan végig kell gondolni, hogy mi a távolabbi cél az adott alkalmazással – csak „házon belüli” használatra szánjuk és nem terjesztjük azt, vagy esetleg a „közjó” érdekében megosztjuk azt másokkal is. Amennyiben egy már meglévő szabad szoftver használata mellett döntünk, akkor a licenc meghatározza szükséges tevékenységünket. Ha viszont mi magunk fejlesztünk vagy

¹⁷ <http://www.386bsd.org/>

¹⁸ <http://www.netbsd.org/>

¹⁹ <http://www.bsdi.com/>

fejlesztetünk ki egy új alkalmazást, érdemes mérlegelni az egyes nyílt forráskódú licencek előnyeit (és esetleges hátrányait), és azok figyelembevételével kiválasztani a megfelelőt.