

Svobodný software

právo na svobodu počítačování

Lukáš Růžička

září 2020

lruzicka@redhat.com

Svoboda



(Freddy-G)

Co je svoboda?

Svoboda je možnost, případně také schopnost, volit, rozhodovat a jednat „podle své vůle“, ať je jakákoli a nést za to přiměřenou odpovědnost.¹

¹<https://cs.wikipedia.org/wiki/Svoboda>

Důležitost svobody

*Ten, kdo se ve jménu
bezpečnosti vzdává svobody,
nezaslouží si ani
svobodu ani bezpečnost.*

(Benjamin Franklin)



Historie



(Bruno Neurath-Wilson)

Historické náhledy na software

1950-60

- software jako předmět **výzkumu**
- software jako součást hardware
- **otevřenost a spolupráce** mezi univerzitami a korporacemi
- **monetizace nebyla** cílem
- často šířen jako **veřejný statek** (public-domain)
- **dostupný zdrojový kód**

Historické náhledy na software

1961-70

- **rozvoj** programovacích jazyků a kompilerů
- **přibalený** (bundled) vs. samostatný software
- rozvoj specializovaného softwaru nezávislého na hardware
- neochota zákazníků **platit** za software **dvakrát**
- soudní spor **USA vs. IBM (1969)** – přibalování software poškozuje konkurenci
- software poskytován s různými **omezeními**

Historické náhledy na software

1971-80

- změna obchodních modelů – např. OS zdarma, ale placené „patche“
- poprvé tzv. **vendor lock-in**
- soudní rozhodnutí z roku 1974 uznalo software za **chránitelný copyrightem** (copyrightable)
- software jako licencovaný produkt
- **konec distribuce zdrojových kódů**, distribuován je pouze tzv. **strojový** (machine) kód
- **patentová ochrana** softwarových řešení

Historické náhledy na software

1981-90

- touha po původní myšlence **sdílení** však nadále existuje mezi **hobíky** a **hackery**
- vznik online sdílených komunit (např. tzv. BBS)
- Richard Stallman a vznik projektu **GNU** (1983)
- **Free Software Manifesto** a vznik **Free Software Foundation** (1985)
- GNU General Public License (1989)²
- jádro projektu GNU – **Hurd**

²Dneska již ve třetím vydání.

Historické náhledy na software

1991-dosud

- jádro **Linux** (na bázi Minix) od Linuse Torvaldse pod licencí GPL
- přidala se **komunita vývojářů** a projekt začal růst
- spojení GNU projektu s linuxovým jádrem – **GNU/Linux**
- Debian – důraz na principy FSF
- některé **části** jádra **nesplňují** definici *free software*
- mnoho dalších distribucí a projektů založených na **GNU/Linux**

Svobodný software

```
31     def __init__(self, path=None):
32         self.file = None
33         self.fingerprints = set()
34         self.logdups = True
35         self.debug = debug
36         self.logger = logging.getLogger(__name__)
37         if path:
38             self.file = open(os.path.join(path, 'seen.txt'), 'a')
39             self.file.seek(0)
40             self.fingerprints.update(self._read())
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getbool('SUPERVISOR_DEBUG')
45         return cls(job_dir(settings), debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         if self.file:
53             self.file.write(fp + os.linesep)
54
55     def request_fingerprint(self, request):
56         return request_fingerprint(request)
```

Co je Svobodný software?

Svobodný software je takový, který zaručuje svým uživatelům **čtyři základní svobody**.

- právo spouštět program, jak si přejete a za jakýmkoliv účelem
- právo zkoumat, jak program funguje, a právo upravit si jej libovolně pro své účely (otevřený kód)
- právo šířit kopie, abyste pomohli ostatním
- právo šířit vámi modifikované verze programu a umožnit komunitě, aby těžila z vašich vylepšení (otevřený kód)

Právo spouštět

- uživatel smí program **spouštět za jakýmkoliv účelem**
- není důležité, za jakým **účelem byl program vytvořen**
- právo spouštět **nezakládá nároky na funkčnost programu**

Právo zkoumat a měnit

- uživatel smí program studovat – k tomu je nutné, aby měl tzv. **otevřený kód** (open source).
- uživatel si smí program upravit pro svou vlastní potřebu, jakožto i pro kohokoliv jiného a **není povinen žádat o svolení**
- uživatel má právo volně spouštět **své vlastní modifikace programu**
- uživatel má právo k programu připojovat další **svobodné knihovny, funkce, a moduly**

Právo šířit

- uživatel má právo **šířit kopie programu zdarma nebo za úplatu** a nemusí nikoho žádat o svolení ani nikomu za toto právo platit
- uživatel má právo program modifikovat **sám pro sebe nebo pro kohokoliv jiného**. Pokud změny zvěřejníte, nejste povinni toto nikomu hlásit
- uživatel má právo **své vlastní modifikace** vydat pod jakoukoliv licencí (program přestane být *free*)
- uživatel má právo distribuovat **zdrojové i binární** kopie programu.

Kopírování zabíjí



Copyright

Copyright je způsobem, jak autorovi zajistit jeho práva k vytvořeným dílům.

- primárně zakazuje komukoliv cokoliv
- použití jenom se svolením autora
- většinou za úplatu (tantiémy)

Copyleft

Copyleft je způsobem, jak zaručit všem svobodu a práva při používání programů.

- není opakem slova **Copyright**, naopak je postaven na **Copyrightu**.
- nemá za účel zpochybnit autorství
- vyžaduje, aby modifikace a redistribuce programu poskytovaly všechna práva, která program poskytoval předtím
- dává volnost programátorům
- chrání programy před **proprietarizací**
- toto je zaručeno správnou licencí

Veřejné použití (public-domain)

Držitelé práv (copyright) dovolují použít svůj kód jakkoliv i bez uvedení jejich autorství.

Volná licence (permissive)

Držitelé práv (copyright) dovolují použít svůj kód jakkoliv, i v proprietárním použití, pokud budou uvedeni dále jako autoři kódu.

Slabý copyleft

Držitelé práv (copyright) sdílejí kód za předpokladu, že lidé, kteří dělají změny do jádra projektu dále předávají toto stejné právo na uživatele.

Silný copyleft

Držitelé práv (copyright) sdílejí kód za předpokladu, že lidé, použijí tento kód v jakémkoliv odvozeném díle, i pouhým nalinkováním, předají toto stejné právo na uživatele.

Licence vyhovující definici svobodného software

- **GNU General Public License (GPL)**
- GNU Lesser General Public License (LGPL)
- Apache License 2.0
- Berkely Database License
- Modified BSD
- Perl 5 License
- X11 License
- a mnoho dalších³

³<https://www.gnu.org/licenses/license-list.html>

Proč některé licence nevyhovují?

Protože chrání **příliš málo** nebo naopak **příliš mnoho**.

Zavřeno!



(David Clode)

Problémy vyplývající z proprietární licence

- náklady na pořízení
- uživatelská podpora
- zabezpečení
- úprava podle přání zákazníka
- vendor lock-in
- možné odchýlení od standardů
- softwarové patenty

Náklady na pořízení

- jednorázové (licence)
- opakované (předplatné)
- kombinované
- riziko počítačového pirátství

Uživatelská podpora

- často zanedbávaná
- mnohdy podfinancovaná
- leckde outsourcing některých úrovní podpory
- obtížně dostupná

Zabezpečení

- neprostá nemožnost zjistit, jak software s bezpečností zachází
- riziko neobjevených chyb
- riziko přeprodávání uživatelských dat třetím stranám
- riziko ovlivňování chování uživatelů

Úprava podle přání zákazníka

neboli **customizace** (customization)

- nelze nechat zákaznický upravit

Vendor lock-in

aneb **závislost zákazníka na produktu** a nemožnost úniku

- firma se vybaví proprietárním řešením
- po čase řešení přestane vychovovat
- data nejsou jednoduše přenositelná jinam
- formát není dobře zdokumentován
- migrace na jiný produkt finančně a časově náročná
- kompatibilita mezi spolupracujícími firmami nejasná
- plánované zastarávání (planned obsolescence)

Standardy

Standardy definují přístup k řešení konkrétních IT problémů.

- kompatibilita napříč počítačovým světem
- svoboda volby

V proprietárních systémech nejsou často respektovány a to způsobuje mnoho problémů.

- nerespektování = konkurenční výhoda
- bludný kruh pro uživatele (vendor lock-in)
- nutnost synchronizace mezi uživateli
- revolvingové náklady na používání

Softwarové patenty

- chrání **duševní vlastnictví** autora
- nutnost zakoupit licenci na jejich použití (drahé)
- ochrana vlastního produktu
- nedostupnost produktu
- zpomalení nebo zastavení vývoje

Příklady proprietárních řešení

Které z následujících jste již viděli naživo?

- Auto s kapotou, která se nedá otevřít jinak než v servisu.
- Mobilní telefon s baterií uvnitř těla a chráněnou plombou.
- Lyže s vázáním, jehož velikost nelze přizpůsobit velikosti boty.
- Kávovar se specifickým tvarem kapslí.
- Písnička ve formátu, který lze přehrát pouze na konkrétním zařízení.

Myslíte, že takové výrobky nám prospívají?

Ze tmy do světla



(Matthew T. Rader)

Výhody sdílení a otevřeného kódu 1

- žádný vendor lock-in
- dodržování průmyslových standardů
- vliv na směřování technologického vývoje
- udržení tempa s konkurencí
- vyšší vliv organizací
- vyšší kvalita kódu
- rychlajší vývoj lepšího softwaru
- vyšší spolehlivost a bezpečí

Výhody sdílení a otevřeného kódu 2

- lepší zachycení potenciálních bezpečnostních hrozeb
- přímá interakce s uživateli
- vývoj a inovace společně s komunitou
- udržení tempa s konkurencí
- rychlejší schopnost reakce na trh
- přilákání a udržení talentů
- uvědomění si změn ve vývoji projektu
- různé pohledy na vývoj
- poskytnutí rámce pro spolupráci

Obchodní modely na bází otevřeného kódu

Podle Wikipedie⁴:

- kód není předmětem prodeje
- prodej uživatelů
- předprodej kódu
- prodej intelektuálního vlastnictví
- prodej proprietárních doplňků
- jiné

⁴https://en.wikipedia.org/wiki/Business_models_for_open-source_software

Kód není předmětem prodeje

- služby
- značkový *merch*
- Software-as-a-Service (SaaS)
- dobrovolné dary
- crowdsourcing

Prodej uživatelů

- partnerství s organizací (Google Summer of Code)
- reklama

Předprodej kódu

- vývoj podle odměny (funkce, které si někdo zaplatí mají přednost)
- placená předobjednávka nebo ukázka (preview)

Prodej duševního vlastnictví

- Dvojí licence neboli *otevřené jádro* (zadarmo pro hobíky, placené pro firmy)
- Prodej certifikací a obchodní značky (trademark)
- Přelicencování na proprietární software (u licencí, které to dovolují)

Prodej doplňků

- proprietární doplňky (moduly, pluginy)
- prodej proprietárních částí softwarového produktu
- prodej proprietárních updatovacích systémů

Pečeme piškoty proprietárně

1. Upečeme piškoty.
2. Zabalíme je.
3. Označíme značkou.
4. Prodáme je.

Pečeme piškoty otevřeně

1. Upečeme piškoty.
2. Zabalíme je.
3. Označíme značkou.
4. **Na obal vytiskneme návod na pečení piškotů, naše poznámky, a nejlepší pečící postupy.**
5. Prodáme je.

Podporujeme 4 základní práva.

1. Použij.
2. Prostuduj.
3. Změň.
4. Sdílej.

Každý peče svoje piškoty

ale pouze za předpokladu, že ...

- dodržuje nařízení o zdraví a bezpečnosti a další zákony
- sdílí svoje úpravy piškotů s komunitou pekařů, chce-li je prodávat
- souhlasí, že komunitu pekařů nebude vinit za jakékoliv následky, které pečení piškotů může mít.

Proč je dobré recept sdílet?

- zlepšování receptu a výroba lepších piškotů v globální spolupráci pekařů
- identifikace možných variací receptu vhodných pro menší trhy, o kterých jste sami nepřemýšleli
- vytvoření komunity pekařů, kteří podporují a šíří váš recept
- objevte talentované pekaře, které můžete zaměstnat ve vašem obchodě
- umístěte se jako čelní výrobce piškotů podle vašeho receptu

Sharing is caring



Někdo nějaké otázky?



Vytvořeno s použitím:

- operačního systému **Fedora Workstation 32**
- programu **T_EXstudio**
- typografického systému **L_AT_EX 2_ε**
- obrázků ze serveru **unsplash.com**

To vše díky free software a ochotě lidí sdílet.

Děkuji za pozornost!