

Miroslav Hrončok, 4.E

Gymnázium Arabská, 2009/2010

Dokumentace k ročníkové práci

# Wine bottle management

<http://sourceforge.net/projects/wibom/>

# Obsah dokumentace

|                                                |    |
|------------------------------------------------|----|
| 1 Úvod                                         | 3  |
| 1.1 Abstrakt                                   | 3  |
| 1.2 Summary                                    | 3  |
| 1.3 Wine                                       | 3  |
| 1.4 Adresářová struktura .wine složky          | 4  |
| 1.5 Práce s lahvemi                            | 5  |
| 2 Zadání                                       | 6  |
| 3 Analýza                                      | 7  |
| 3.1 Rozhlédnutí                                | 7  |
| 3.1.1 Wine-doors                               | 7  |
| 3.1.2 CrossOver                                | 7  |
| 3.1.3 PlayOnLinux                              | 7  |
| 3.2 Formální zadání                            | 8  |
| 4 Návrh                                        | 8  |
| 5 Implementace                                 | 9  |
| 5.1 Implementace jádra                         | 9  |
| 5.1.1 Jazyk                                    | 9  |
| 5.1.2 Kam s lahvemi?                           | 9  |
| 5.1.3 Výchozí lahev                            | 9  |
| 5.1.4 Nová lahev a kopie lahve                 | 9  |
| 5.1.5 Import lahve                             | 9  |
| 5.1.6 Mazání lahve                             | 10 |
| 5.1.7 Spustit software v lahvi                 | 10 |
| 5.2 Implementace řádkového rozhraní            | 10 |
| 5.3 Implementace grafického rozhraní           | 10 |
| 5.3.1 Grafické rozhraní                        | 10 |
| 5.3.2 Jazyk                                    | 10 |
| 5.3.3 Nemám elementární představu, co dál psát | 10 |
| 6 Testování                                    | 11 |
| 7 Výsledky                                     | 11 |
| 8 Závěr                                        | 11 |
| 8.1 Licence                                    | 11 |
| 9 Seznam literatury                            | 13 |
| 10 Abecední seznam zkratk                      | 13 |
| 11 Přílohy                                     | 14 |
| 11.1 Leaflet                                   | 14 |
| 11.2 Instalační příručka                       | 14 |
| 11.2.1 Požadavky                               | 14 |
| 11.2.2 Instalace                               | 14 |
| 11.3 Uživatelská příručka                      | 15 |
| 11.4 Zdrojový kód a program                    | 15 |

# 1 Úvod

## 1.1 Abstrakt

Aplikace, nazvaná **Wine bottle management**, slouží, jak název napovídá, ke správě tzv. *Wine lahví*.

Wine je program, který slouží k běhu aplikací určených pro Microsoft Windows na jiných operačních systémech, především pak na GNU/Linuxu. Tento program vytváří v domovském adresáři uživatele složku `.wine`, která obsahuje kromě uživatelských nastavení také zjednodušenou podobu Windows registrů a adresářovou strukturu simulovaného disku C:. Mnoho uživatelů Wine však netuší, že mohou těchto složek mít více. Ty se pak nazývají bottles neboli lahve. Díky lahvím můžete mít na jednom systému několik navzájem izolovaných instancí „Pseudowindows“.

Samotný program Wine však nepřináší žádný systém na správu těchto složek a k jejich obsluze je třeba znát základy práce s Bashem, případně jiným systémovým shellem, a princip fungování Wine. Tímto systémem je právě **Wine bottle management**.

Správce Wine lahví umožňuje jednoduše spravovat jednotlivé lahve, vyvářet nové, klonovat je, nastavovat je, instalovat do nich aplikace pro Windows a to všechno v jednoduchém grafickém rozhraní.

## 1.2 Summary

Application called **Wine bottle management** is used to (as its name suggests) manage so-called *Wine bottles*.

Wine is the program for running applications for Microsoft Windows on other operating systems, primary on GNU/Linux. This program creates in the user's home directory a folder called `.wine` containing user preferences, the simplified version of Windows registers and files structure of simulated disk C:. Many Wine users don't know there can be more those folders coexisting. They are called bottles. Because of bottles it is possible to have a few isolated instances of "Pseudowindows" on one system.

Wine offers no system for managing those folders. It is needed to have a basic knowledge of using Bash or any other system shell and knowledge of principles of Wine's job to be able to use bottles. And this is business of **Wine bottle management**.

Wine bottles can be easily managed, they could be created, cloned. Windows software can be easily installed into them. Everything is possible through easy GTK interface.

## 1.3 Wine

*Wine je software vytvářející aplikační rozhraní, které umožňuje chod aplikací pro Microsoft Windows pod jinými (převážně unixovými) operačními systémy. Použitelnost se liší v závislosti na povaze software, nejlepší výsledky lze očekávat u aplikací psaných pro Windows 95, případně Windows 98. Wine vytvořili v roce 1993 Bob*

*Wine je akronym pro **Wine is not emulator**, což v češtině znamená Wine není emulátor. Tato věta záměrně upozorňuje na to, že Wine je pouze soustava API, a proto neumí spouštět Windows aplikace na jiné architektuře, než je x86.*

*Wine je svobodný software uvolněný pod licencí LGPL. Projekt byl původně distribuován pod licencí MIT, ale kvůli obavám z komerčních verzí, které by svá vylepšení neposkytovaly zpět do Wine, byla od března 2002 licence změněna na LGPL.*

*První betaverze Wine (verze 0.9) byla uvolněna 25. října 2005 po 12 letech vývoje. Verze 1.0 byla uvolněna 17. června 2008 po 15 letech vývoje. [cj9h9]*

#### 1.4 Adresářová struktura .wine složky

Po prvním spuštění programu Wine je v domovském adresáři uživatele, který ho spustil, automaticky vytvořena složka .wine. V této části dokumentace se budu věnovat její struktuře.

Tato složka, respektive jakákoli lahev, představuje izolovanou instanci mini-kopie operačního systému Windows. Obsahuje dvě složky a tři textové soubory.

Složka `dosdevices` obsahuje symbolické odkazy na adresáře, jejichž obsah má být podstrčen Windows programům jako obsah konkrétních disků. Ve standardní nově vytvořené lahvi obsahuje pouze dva smlinky - `z:`, který odkazuje na kořenový adresář souborového systému reálného operačního systému, na kterém Wine běží, a `c:`, který odkazuje na druhou složku v lahvi.

Složka `drive_c`, na kterou zmíněný odkaz vede, obsahuje adresářovou strukturu disku `C:` obecného systému Windows. Obsahuje složky `Program Files`, `Windows` a `Users`, které slouží k tomu, aby se aplikace určené pro Windows měli kam nainstalovat (ve smyslu rozbalení).

Textové soubory v lahvi slouží jako Windows registry, aby se aplikace měli kam zaregistrovat.

**Poznámka:** *Nepochopil jsem, proč je nutné po rozbalení binárních souborů s programem ještě zasahovat do tak zvrácené struktury, jako je Windows registr, ale ve Wine je reprezentován textovým souborem a to je mi bližší.*

Soubor `system.reg` uchovává systémová nastavení Windows, soubor `user.reg` nastavení konkrétního uživatele Windows. Wine v tomto však počítá pouze s jedním uživatelem, protože další uživatel reálného operačního systému má své vlastní lahve. Poslední soubor `userdef.reg` obsahuje určité cesty k uživatelským adresářům, jako je `My Pictures`. Každý vede do složky `Users` na fiktivním disku `C:`, kde jsou však jenom ukazatele na reálné umístění (většinou přímo na domovský adresář uživatele reálného OS).

Nová (prázdná<sup>1</sup>) lahev má velikost asi 7 MB. Určitým ručním probráním nepotřebných součástí se její velikost dá zmenšit asi na 5 MB. V závislosti na instalovaném softwaru její velikost logicky stoupá.

## 1.5 Práce s lahvemi

Berme nyní v úvahu, že neznáme pojem lahev, respektive bottle, a nevíme, že se něco takového dá ve Wine použít.

Pokud bychom vytvořili novou .wine složku a nainstalovali pomocí Wine například aplikaci Internet Explorer 7 nemohli bychom zároveň nainstalovat Internet Explorer 6. Pokud bychom o něco takového stáli, například kvůli testování kompatibility námi vytvořeného webu, museli bychom hledat řešení.

Jedním z nich by bylo přejmenování složky .wine například na .wine-ie7 tak, aby Wine při příštím spuštění vytvořilo novou složku .wine, do které bychom mohli nainstalovat Internet Explorer 6. Složky bychom mohli střídavě přejmenovávat a podsouvat je tak aplikaci Wine v závislosti na tom, kterou verzi prohlížeče chceme spustit. Nevědomky bychom tak vytvořili dvě lahve. Dvě izolovaná prostředí Windows, respektive Wine.

Tento způsob je ale zbytečně složitý. Existuje elegantnější varianta. Totiž taková, že nebudeme s adresáři nijak manipulovat, ale aplikaci Wine prostě sdělíme, který adresář má použít. Pokud neexistuje, vytvoří v něm novou prázdnou lahev.

*(...) To můžu udělat dvěma způsoby:*

*Bud' dvouřádkově (v terminálu):*

```
export WINEPREFIX="$HOME/.wine-heroes3"  
winecfg
```

*Nebo jednořádkově (klidně přes dialogové okno spustit):*

```
env WINEPREFIX="$HOME/.wine-heroes3" winecfg
```

*V okně s nastavením Wine ještě doporučuji prokliknout záložku **Zvuk**, aby bezpečně fungoval zvuk. Ted' už jen nainstalovat Heroesy do nové bottle. A jak? Před každým příkazem použijí určení .wine složky, stejně jako před chvílí. A pak už jen spouštěč na Heroes 3:*

```
#!/bin/bash  
cd ~/.wine-heroes3/drive_c/Program\ Files/3D0/Heroes\ 3\ Complete  
export WINEPREFIX="$HOME/.wine-heroes3"  
wine explorer /desktop=HoMM3,800x600 ./HEROES3.EXE
```

*Na podobném principu funguje i skript ies4linux, používá pro každou verzi Internet Exploreru jinou bottle, protože každá lahev*

---

<sup>1</sup> Ve smyslu softwaru. Pokud by byla opravdu prázdná, neměla by 7 MB a nebyla by nadále bottle, ale pouze prázdným adresářem. Pokud si tedy kladete otázku, jak může prázdná bottle zabírat na disku určitou velikost, vězte, že prázdná láhev od vína má také určitou hmotnost.

*představuje jednu pseudoinstalaci Windows s registry i se složkou, která představuje disk C:.* [dnuv2]

## 2 Zadání

Mým zadáním je vytvořit aplikaci, která bude umožňovat uživateli bez speciálních znalostí Wine nebo Bashe spravovat lahve, ovladatelnou pro začátečníka. Měla by mít následující funkce:

- vytváření a mazání lahví
- import existujících lahví
- výchozí prázdná lahev, ze které se budou vyrábět nové
- duplikování lahví
- instalace softwaru do lahve
- možnost konfigurace Wine a registrů pro každou lahev

Všechny tyto funkce budou ovladatelné pomocí řádkových příkazů a k nim bude vytvořeno samostatné grafické rozhraní, které může mít některé funkce navíc.

## 3 Analýza

### 3.1 Rozhlédnutí

V současné době existuje několik nadstaveb nad Wine. Některé z nich již nejsou vyvíjeny a jsou nekompatibilní se současnými verzemi Wine. Jiné jsou komerční a nemohou být například obsaženy v linuxových distribucích.

#### 3.1.1 Wine-doors

Open-source projekt, který funguje jako balíčkovací systém s Windows aplikacemi. Nepodporuje však práci s lahvemi. Cíle Wine-doors a Wine bottle managementu se nepřekrývají, ačkoli pro současné využití obou aplikací tak, aby spolu mohli kooperovat, by bylo potřeba obě aplikace mírně přizpůsobit.

#### 3.1.2 CrossOver

*Vylepšená komerční verze Wine. Nazývá se také CrossOver Office nebo cxoffice. CrossOver (k dispozici také pro Mac OS X) vyvíjí a prodává firma CodeWeavers. Kód je založen na Wine, přičemž CodeWeavers poskytuje (na základě licencí GPL a LGPL) zdrojové kódy své upravené verze volně k dispozici a přispívá také k vývoji Wine.*

*Hlavní výhoda oproti Wine je u CrossOver v tom, že se jedná o vyladěný produkt, který podporuje stabilní spouštění velkého množství windowsových programů, s nimiž má zatím Wine potíže (kancelářské balíky, pluginy do webových prohlížečů, hry, ...). Instalace je jednoduchá, není nutná žádná kompilace a přidávání a odebrání nových programů je velmi snadné. Kromě toho získává zákazník zakoupením CrossOver oficiální podporu. CrossOver je možné si zdarma stáhnout a vyzkoušet.*

*Je možné si zakoupit tři různé verze: Standard, Professional a Server (umožňující spouštění aplikací pro tenké klienty na serveru). [cj9t2]*

CrossOver někdy vytváří lahev, ale většinou jen jednu, aby nepoškodil standardní .wine složku. Je možné stáhnout si zkušební verzi CrossOveru a nainstalovat s její pomocí nějaký software, který se do Wine normálně instaluje velice složitě. Poté CrossOver lahev někam zálohovat a po odebrání zkušební verze ji naimportovat do Wine bottle managementu. Není to v rozporu se zákonem, ani licencí. Otázku morálky nechávám na uživateli.

#### 3.1.3 PlayOnLinux

*PlayOnLinux je jednoduchá aplikace vytvořená v Pythonu. (...) Nevýhodou může být větší prostorová náročnost, PlayOnLinux si totiž pro každou aplikaci vytváří v adresáři ~/.PlayOnLinux vlastní Wine prefix<sup>2</sup> s vlastní konfigurací, registry atd. Pokud chcete přes PlayOnLinux nainstalovat a používat aplikaci, která není oficiálně*

---

<sup>2</sup> Což je jiné označení pro lahev.

*podporována, můžete si podle návodu napsat vlastní instalační skripty v Bashi.*

*PlayOnLinux také podporuje pluginy, například pro podporu DosBoxu, pro benchmark grafické karty a další. [dn899]*

Nadstavba PlayOnLinux sice využívá lahve, ale neumožňuje s nimi pracovat na jiné úrovni, než podle vztahu jedna aplikace ~ jedna lahev, což neumožňuje tvořit speciální lahve, například pro hraní, školu a podobně.

### 3.2 Formální zadání

Program Wine bottle management při prvním spuštění vytvoří výchozí Wine lahev. Uživatel může vytvářet nové lahve - klonováním výchozí lahve. Případně lze klonovat i jiné než výchozí lahve. Pokud uživatel vytvořil nějaké lahve jinak než pomocí této aplikace, může je importovat do správy aplikace. Libovolnou lahev kromě výchozí je možné smazat - soubory se přesunou do koše, takže je možné je obnovit a lahev importovat zpět.

S lahvemi lze také pracovat. Je možné otevřít složku s lahví ve výchozím správci souborů, konfigurovat lahev pomocí nástroje konfigurace Wine a editovat registry Windows.

Do lahví lze přidávat software a odebírat ho pomocí nástroje **Přidat/Odebrat software**, který nabízí samotné Wine. Kromě toho je možné „v lahvi“ spustit jakýkoli program, který Wine podporuje.

## 4 Návrh

Aplikace se bude skládat ze tří součástí:

- **jádro** - sada skriptů, které budou obstarávat veškeré práce s lahvemi, uživatel je nebude spouštět
- **řádkové rozhraní** - rozhraní pro příkazovou řádku, které bude sdružovat práci se skripty
- **grafické rozhraní** - primární rozhraní pro běžné uživatele, kteří budou spravovat lahve především klikáním



# 5 Implementace

Implementaci jsem rozdělil na tři části:

## 5.1 Implementace jádra

Jádro aplikace tvoří sada několika skriptů, které reprezentují jednotlivé funkce programu.

### 5.1.1 Jazyk

Jelikož práce s lahvemi je především o proměnných Bashe, parametrech a zpracovávání složek a úpravách textových souborů, rozhodl jsem se jádro napsat v Bashi samotném. Ten mi umožní pracovat efektivně, navíc je obsažen ve většině linuxových distribucí i v jiných UNIX systémech, na kterých běží program Wine.

### 5.1.2 Kam s lahvemi?

Dříve, než se začnu zabírat jednotlivými operacemi, jako je tvorba nové lahve a podobně, musím vytvořit výchozí místo pro lahve a nějakým způsobem je evidovat.

Z principu mohou být jednotlivé lahve rozeseté po celém souborovém systému operačního systému (včetně těch virtuálních). Přesto je vhodné mít výchozí adresář pro lahve, aby v nich mohl být pořádek. Jako tento adresář jsem zvolil složku `~/.local/share/bottles`, jelikož to podporuje určitou logiku ukládání podobných dat aplikací v domovském adresáři.

Při prvním spuštění aplikace se tedy tento adresář musí vytvořit, stejně tak, jako se v něm vytvoří seznam lahví a výchozí lahev (viz dále).

Seznam lahví je reprezentován textovým souborem `bottles.lst` v tomtéž adresáři. Tento soubor má velice jednoduchou strukturu – totiž jeden řádek je jedna absolutní cesta k lahvi.

### 5.1.3 Výchozí lahev

Výchozí lahev se nachází v adresáři `default` ve složce s lahvemi. Pokud aplikace zjistí, že tomu tak není (první spuštění programu, účelově i omylem smazaná výchozí lahev), vytvoří ji jako novou `.wine` složku. Výchozí lahev není zapsána v seznamu lahví, program s ní počítá automaticky.

### 5.1.4 Nová lahev a kopie lahve

Vytvoření nové lahve pak probíhá jako vytvoření kopie výchozí lahve. Vytvoření kopie jakékoli lahve se skládá ze dvou kroků:

1. Kopírování souborů
2. Vytvoření nového záznamu v seznamu lahví

### 5.1.5 Import lahve

Další z funkcí programu je import existující lahve. Pokud tedy zadáme skriptu nějaký adresář, skript vyhodnotí, jestli se jedná o lahev a jestli se nesnažíme importovat lahev, která je na seznamu (nebo výchozí lahev). Pokud je vše

v pořádku, adresář je přidán do seznamu, ale není nikam fyzicky kopírován, nebo přesouván.

### 5.1.6 Mazání lahve

Může nastat situace, že uživatel chce nějakou lahev smazat. V takovém případě se řádek s lahví odstraní ze seznamu a soubory lahve se přesunou do koše podle specifikace FreeDesktop.org díky utilitě `trash-cli` [dzpb6]. Pokud tato utilitka není přítomna, soubory jsou smazány.

### 5.1.7 Spustit software v lahvi

Jedním z nejpodstatnějších skriptů je skript na spuštění nějakého programu v lahvi. Ten ověří, jestli je zadaným parametrem lahev a pokud ano, pokusí se v ní spustit uživatelem zvolený program - ať už Wine, nebo jinou aplikaci. Samozřejmostí je předávání exitkódu.

## 5.2 Implementace řádkového rozhraní

Řádkové rozhraní jen sdružuje práci se skripty do jediného příkazu `wibom`. Je k němu dostupná manuálová stránka, kterou po nainstalování zobrazíte příkazem `man wibom`.

## 5.3 Implementace grafického rozhraní

### 5.3.1 Grafické rozhraní

Rozhodl jsem se využít grafických knihoven, které již existují. Pro tři nejpoužívanější desktopové grafické prostředí v Linuxu (GNOME, KDE a Xfce), existovaly dvě volby: GTK+ a Qt.

Jelikož počet uživatelů GNOME (knihovny GTK+) a KDE (knihovny Qt) je srovnatelný, rozhodl jsem se pro knihovny GTK+, protože ty používá i třetí nejpoužívanější prostředí, totiž Xfce.

### 5.3.2 Jazyk

Přestože jádro aplikace je napsané v Bashi, psát grafickou aplikaci v tomto jazyce není příliš komfortní. Zvolil jsem proto Ruby, které mi poskytne komfort skriptovacího jazyka a s GTK+ objekty se v něm pracuje podobně jako v jazyce C.

### 5.3.3 Nemám elementární představu, co dál psát

## 6 Testování

Funguje to.

## 7 Výsledky

Funguje to tak, jak jsem chtěl.

## 8 Závěr

Funguje to tak, jak by mělo, aby to fungovalo tak, jak jsem chtěl.

### 8.1 Licence

Tento software je vydán pod novou BSD licencí, jejíž neoficiální český překlad zní:

*Copyright © 2009, Miro Hrončok [http://hroncok.cz]. Všechna práva vyhrazena.*

*Redistribuce a použití zdrojových i binárních forem díla, v původním i upravovaném tvaru, jsou povoleny za následujících podmínek:*

- *Šířený zdrojový kód musí obsahovat výše uvedenou informaci o copyrightu, tento seznam podmínek a níže uvedené zřeknutí se odpovědnosti.*
- *Šířený binární tvar musí nést výše uvedenou informaci o copyrightu, tento seznam podmínek a níže uvedené zřeknutí se odpovědnosti ve své dokumentaci a/nebo dalších poskytovaných materiálech.*
- *Ani jméno vlastníka práv, ani jména přispěvatelů nemohou být použita při podpoře nebo právních aktech souvisejících s produkty odvozenými z tohoto software bez výslovného písemného povolení.*

**TENTO SOFTWARE JE POSKYTOVÁN DRŽITELEM LICENCE A JEHO PŘÍSPĚVATELI „JAK STOJÍ A LEŽÍ“ A JAKÉKOLIV VÝSLOVNÉ NEBO PŘEDPOKLÁDANÉ ZÁRUKY VČETNĚ, ALE NEJEN, PŘEDPOKLÁDANÝCH OBCHODNÍCH ZÁRUK A ZÁRUKY VHODNOSTI PRO JAKÝKOLIV ÚČEL JSOU POPŘENY. DRŽITEL, ANI PŘÍSPĚVATELÉ NEBUDOU V ŽÁDNÉM PŘÍPADĚ ODPOVĚDNI ZA JAKÉKOLIV PŘÍMÉ, NEPŘÍMÉ, NÁHODNÉ, ZVLÁŠTNÍ, PŘÍKLADNÉ NEBO VYPLÝVAJÍCÍ ŠKODY (VČETNĚ, ALE NEJEN, ŠKOD VZNIKLYCH NARUŠENÍM DODÁVEK ZBOŽÍ NEBO SLUŽEB; ZTRÁTOU POUŽITELNOSTI, DAT NEBO ZISKŮ; NEBO PŘERUŠENÍM OBCHODNÍ ČINNOSTI) JAKKOLIV ZPŮSOBENÉ NA ZÁKLADĚ JAKÉKOLIV TEORIE O ZODPOVĚDNOSTI, AŤ UŽ**

*PLYNOUCÍ Z JINÉHO SMLUVNÍHO VZTAHU, URČITÉ ZODPOVĚDNOSTI NEBO PŘEČINU (VČETNĚ NEDBALOSTI) NA JAKÉMKOLIV ZPŮSOBU POUŽITÍ TOHOTO SOFTWARE, I V PŘÍPADĚ, ŽE DRŽITEL PRÁV BYL UPOZORNĚN NA MOŽNOST TAKOVÝCH ŠKOD. [cj9s5]*

## 9 Seznam literatury

Kód v hranaté závorce nejen identifikuje citaci v textu dokumentace, ale zároveň slouží jako zkrácený odkaz na dílo, ze kterého cituji (pokud je dostupné na Internetu), nebo na informace o něm (pokud není). Před kód je nutné napsat adresu <http://jdem.cz/> - například tedy <http://jdem.cz/cj9h9>. Takový odkaz pak vede na stejnou adresu, která je uvedena u citace.

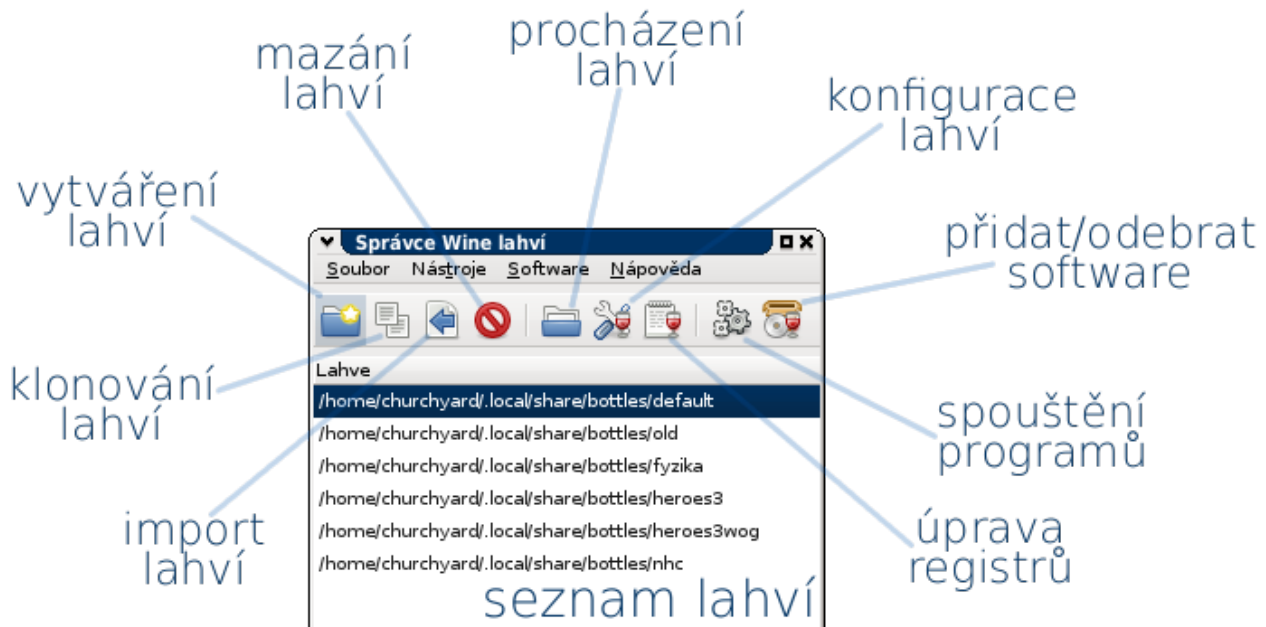
- [cj9h9] Adam Živněř, **Wine** na české Wikipedii,  
<http://cs.wikipedia.org/wiki/Wine>
- [cj9s5] různí překladatelé, **BSD licence** na české Wikipedii,  
[http://cs.wikipedia.org/wiki/BSD\\_license](http://cs.wikipedia.org/wiki/BSD_license)
- [cj9t2] Robert Krátký, **CrossOver** v databázi softwaru na AbcLinuxu.cz,  
<http://www.abclinuxu.cz/software/system/emulatory/crossover>
- [dnuv2] Miroslav Hrončok, **Vinný sklípek, aneb co je to bottle?**  
<http://www.linuxexpres.cz/blog/vinny-sklipek-aneb-co-je-to-bottle>
- [dn899] Dan Vrátil, **Wine a jeho pomocníci**,  
<http://www.abclinuxu.cz/clanky/recenze/wine-a-jeho-pomocnici>
- [dzpb6] Program **trash-cli** uvolněný pod licencí GPLv2,  
<http://code.google.com/p/trash-cli/>

## 10 Abecední seznam zkratk

- **GNOME** - *GNU Network Object Model Environment*, desktopové prostředí, které používá knihovny GTK+, <http://www.gnome.org>
- **GNU** - rekurzivní zkratka *GNU's Not Unix*, <http://www.gnu.org>
- **GPLv2** - GNU General Public License, version 2, <http://jdem.cz/dzpe5>
- **GTK+** - dříve zkratka *GIMP Toolkit*, nyní se spíše používá *The GTK+ Project* bez vysvětlení zkratky, knihovna pro tvorbu grafických aplikací, <http://www.gtk.org>
- **KDE** - dříve *Kool Desktop Environment*, dnes jen *K Desktop Environment*, desktopové prostředí, které používá knihovny Qt, <http://www.kde.org>
- **LXDE** - Další desktopové prostředí používající knihovny GTK+, *Lightweight X11 Desktop Environment*, <http://www.lxde.org>
- **Qt** - další knihovna pro tvorbu grafických aplikací, nejedná se vlastně o zkratku, <http://qt.nokia.com>
- **Xfce** - dříve *XForms Common Environment*, dnes neznamena nic, třetí nejpoužívanější desktopové prostředí, používá knihovny GTK+, <http://www.xfce.org>

# 11 Přílohy

## 11.1 Leaflet



Hezčí varianta je přiložena, toto je pouze ukázka.

## 11.2 Instalační příručka

### 11.2.1 Požadavky

- UNIXový operační systém, například Debian GNU/Linux
- Bash
- X server
- Interpret jazyka Ruby (ruby)
- Knihovnu GTK+ (libgtk2)
- Ruby-GNOME2 (libgnome2-ruby)
- Ruby-GetText (libgettext-ruby)
- Utilita Trash (trash-cli)
- XDG Utils (xdg-utils)
- Výchozí nouzové téma ikon pro FreeDesktop.org (hicolor-icon-theme)
- Wine

### 11.2.2 Instalace

Po nainstalování všech požadavků stačí nakopírovat obsah adresáře `usr` na přiloženém médiu do systémového adresáře `/usr`. Program spustíte příkazem

wibom-gtk a pokud používáte nějaké desktopové prostředí, například GNOME, Xfce nebo LXDE, najdete jej i v nabídce aplikací v kategorii Příslušenství.

### 11.3 Uživatelská příručka

Bohužel ještě není dokončena, bude součástí programu.

### 11.4 Zdrojový kód a program

Obojí je totožné, protože jsem zvolil skriptovací jazyky Bash a Ruby. Program a tedy i zdrojové kódy naleznete na přiloženém médiu.