Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

Vytvoření programu pro editaci registrů Windows pomocí Linuxu

Bakalářská práce

Martin Klíma

Vedoucí: Mgr. Pech Jiří Ph.D.

České Budějovice 2014

Klíma M., 2014: Vytvoření programu pro editaci registrů Windows pomocí Linuxu. [Creating application for editing Windows Registry using platform Linux. Bc. Thesis, in Czech.] – 42 p., Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

Anotace:

Bakalářská práce se zaměřuje na zveřejněné poznatky o struktuře Windows Registrů. Je zde podrobně popsána logická struktura a fyzická struktura Windows Registrů. Dále se práce zaměřuje na dostupné open-source nástroje pro editaci Windows Registrů. V závěru práce je popsán postup, pro vytvoření aplikace, která umožňuje přistupovat k Windows Registrům z systému Linux pomocí grafického prostředí - GUI.

Annonation:

This Bachelor thesis is focused on public knowledge about internal structure of Windows Registry. In this thesis is described in deep logical and physical structure of Windows Registry. After that is there list of open-source tools for accessing Windows Registry. In end of this work is described process how to create application which allow users to access Windows Registry from Linux with graphical interface – GUI.

Prohlašuji, že svoji bakalářskou práci jsem vypracoval/a samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce.

Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce.

Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Rostocku dne 12. března 2014

_ _ _ _ _ _ _ _ _ _ _ _ _

Poděkování

Rád bych poděkoval svému školiteli Mgr. Jiřímu Pechovi Ph.D. za jeho pomoc při tvorbě této práce.

Rád bych poděkoval Timothy D. Morganovi za jeho práci na publikaci „The Windows NT Registry File Format“. Rád bych poděkoval Richard W.M. Jonesovi za jeho práci na knihovně HIVEX.

Rád bych poděkoval Endy, Ditě, Žofákovi, Evičce († 2009) a Líze († 26.11.2013) za jejich neutuchající hravost a dobrou náladu, která pomohla ke vzniku této práce.

A jako poslední bych rád poděkoval svoji přítelkyni – Zuzce Kotrbové, za její starostlivost, milý úsměv a péči, kterou mi věnovala.

Obsah

Seznam Obrázků

1 Úvod

Když jsem se rozhodl opustit operační systém Windows a vyzkoušet jiný systém byl to celkem šok. Všechny aplikace, které jsem dřív denně používal nyní nefungovaly a bylo nutné za ně najít náhradu.

Některé se daly nahradit snadno – jako například Microsoft Office pomocí Libre Office ale bylo tu velké množství aplikací, které se nahradit nedaly. Případně, jejich náhrady byly velice složité a mnohdy se jedno pouze o command-line aplikace, které jsem ani nevěděl, jak správně použít.

Pokud chceme docílit toho, aby byl Linux brán jako skutečná alternativa k operačnímu systémů Windows a přimět uživatele k jeho většímu užívání, musíme jim dát alternativy k aplikacím, na které byli zvyklí z minulého systému. A ne pouze k funkcím těchto aplikacích ale také k rozhraní, na které byli navyklí a uměli ho používat z předchozího systému.

Z toho důvodu jsem si vybral tuto bakalářskou práci. Chci dát uživatelům možnost mít stejný komfort pro manipulaci s Windows Registry v Linuxu jako ve Windows.

2 Cíle práce

V této bakalářské práce se chci zaměřit na tyto cíle:

1. Zmapovat a popsat zveřejněné informace o struktuře Windows Registrů.
2. Popsat již vytvořené alternativní nástroje pro přístup k Windows Registrům.
3. Vytvořit aplikaci, která bude splňovat následující:
   a) Bude mít grafické uživatelské rozhraní. (GUI)
   b) Bude umožňovat číst, měnit, vytvářet a mazat klíče.
   c) Její funkčnost bude otestovaná na ukázkových datech různých verzí systému Windows.

3 Windows Registry

3.1 Co jsou to Windows Registry?

Každý operační systém potřebuje nějak ukládat informace. Ať už to jsou systémové informace, bezpečnostní informace, hardwarové profily či údaje o uživatelích. Také každá trošku pokročilá klientská aplikace musí někam ukládat data o svém nastavení a opět je tam při spouštění najít. Bez těchto dat by byla většina aplikací prakticky nepoužitelná.

Proto je potřebné mít v operačním systému určitý způsob jak tyto informace uchovávat a také zajistit jejich přístupnost všem aplikacím, které je potřebují a také dostatečnou bezpečnost a stabilitu, aby se tato data nemohla ztratit a nebo poškodit. Z těchto důvodů přišel Microsoft s centrální databází, které dal název Microsoft Registry.

3.2 Vznik Windows Registrů

Nejstarší systém MS-DOS používal pro uchovávání systémových informací soubor Config.sys [Honeycutt, 2005] S příchodem Windows 3.0 představil Microsoft koncept textových konfiguračních souborů - takzvané .ini soubory.

Tyto textové soubory mohli uchovávat více nastavení v jednom souboru oddělených do jednotlivých sekcí. Jak vypadala struktura těchto .ini souborů můžete vidět zde:
Obrázek 1: Ukázka souboru SYSTEM.INI

Ukládání nastavení do textových souborů se neukázalo jako velmi praktický způsob. Každá aplikace si ukládala svoje nastavení do jiné složky a bylo těžké dohledat konkrétní konfigurační soubor pro danou aplikaci.

Stejně tak nevyhovoval textový formát. Do textového formátu se těžko ukládala binární data (bylo nutné je složitě kodovat) a také nebylo žádné propojení mezi jednotlivými daty.

Dalším důvodem, proč Windows textové soubory nestačily, byla vysoká rizikovost smazání souboru a tím ztracení celé konfigurace bez možnosti obnovení. Jak tento způsob řeší Windows Registry popisuji v kapitole „Bezpečnost & Atomicita“.

```
; for 16-bit app support
[drivers]
wave=mmdrv.dll
timer=timer.drv
[mci]
[driver32]
[386enh]
woafont=dosapp.FON
EGA80WOA.FON=EGA80WOA.FON
EGA40WOA.FON=EGA40WOA.FON
CGA80WOA.FON=CGA80WOA.FON
CGA40WOA.FON=CGA40WOA.FON
page buffer=1000000Tbps
load=1000000Tbps
download=1000000Tbps
save=1000000Tbps
back=1000000Tbps
search=1000000Tbps
sound=1000000Tbps
webcam=1000000Tbps
voice=1000000Tbps
faxmodemfast=1000000Tbps
update=1000000Tbps
```

Obrázek 1: Ukázka souboru SYSTEM.INI

Windows 3.1 představily takzvané OLE (object linking and embending), což byl přímý předchůdce Windows Registrů. Samotné Windows Registry jako centrální databáze se objevily až s příchodem systému Windows 95 a Windows NT.

V současné době Windows Registry představují skutečně kritickou část operačního systému Windows.
Při každém spuštění systému se z nich načítají ovladače, systémové nastavení, informace o politice zabezpečení, uživatelské účty, hardwarove informace, síťová nastavení a mnoho dalšího. Stejně tak instalované aplikace tyto Windows Registry hojně využívají a ukládají do nich informace a data potřebná k jejich provozu a správnému fungování. [Kokoreva, 2004]

3.3 Bezpečnost & Atomicita

Windows Registry obsahují všechna nastavení, které jsme použili v operačním systému a stejně tak v dalších aplikacích, které na Windows používáme. Během každé operace ve Windows se k nim přistupuje a čtou se z nich hodnoty. Dle některých dat až 2000 operací při bootování systému. [Honeycutt, 2005] Kdykoliv se něco s Windows Registry pokazí, způsobí to pád celého systému a systém se musí přeinstalovat.

Proto je pochopitelné, že se Microsoft snaží omezit zásahy zvenčí, které by způsobily kolaps celých Windows Registrů a také používá mechanizmy, které mají zabezpečit stabilitu a atomicitu dat. [Microsoft, 2010]

Mezi tyto mechanizmy patří:

- Odepření běžným uživatelům přístup k hive souborům s Windows Registry. Pro manipulaci s těmito soubory, musíte být přihlášen jako Administrator.
- Ukládání veškerých transakcí do logu, z kterého je později možnost obnovit původní data. [Norris, 2009]
- Vytváření alternativních kopií, pro pozdější kompletní obnovu Windows Registrů.
- Kontrolní součet každé hive v hlavičce, který indikuje zda není registr porušen.
- „Hive Flush“ - procházení veškerých hive v systému a automatická oprava nebo smazání vadného sektoru – jak uvádí v Windows Kernel Internals NT Registry Implementation [Probert, nedatovano]

3.4 Uzavřenost

Pokud se podíváme na předchozí chování firmy Microsoft, která se snažila svoje technologie držet pod-pokličkou, určitě vás nepřekvapí, že ani k Windows Registrům, neposkytuje žádnou kompletní dokumentaci k tomu, jak jsou tyto data ukládána v binární podobě a jak s nimi dále manipulovat.

Veškeré zdroje z oficiálních kanálů odkazují buď na Regedit32 - tedy oficiální nástroj, který Microsoft doporučuje pro manipulaci s Windows Registry a nebo na oficiální Windows API, přes kterou lze k Windows Registrům přistupovat jako ke službě. Taktéž veškerá dokumentace, kterou lze dohledat přímo od Microsoftu se zaobírá Windows

API pro přístup k Windows Registrů, případně vysvětlováním účelu jednotlivých klíčů a nastavením, která ovlivňují chování systému Windows.

Můžeme se zde domnívat, že to může být snaha omezit nechtěné změny v datech, která jsou tak důležitá pro samotnou funkci systému a jeho bezpečnost, a nebo zabránit útokům z vnější, které by mohli útočníkovi přinést skutečně citlivé informace jako hesla a osobní údaje uživatele. Všechna tato rizika jsou opodstatněná ale nutno dodat, že pokud již útočník získá přístup k tomuto souboru, není nic jednodušího než použit opět Windows API z jiného stroje, na kterém jsou Windows instalovány a díky tomu se k těmto datům dostat a snadno je přečíst nebo změnit.

Jediná možnost, která v současné době existuje pro přenos dat z Windows Registrů do jiného čitelného formátu, je exportování klíčů do formátu .reg, který je čistě textový. Ale opět je to možné, pouze za použití Windows API a tedy stanice, která musí mít Windows instalovány.

4 Open-source nástroje pro práci s Windows Registry

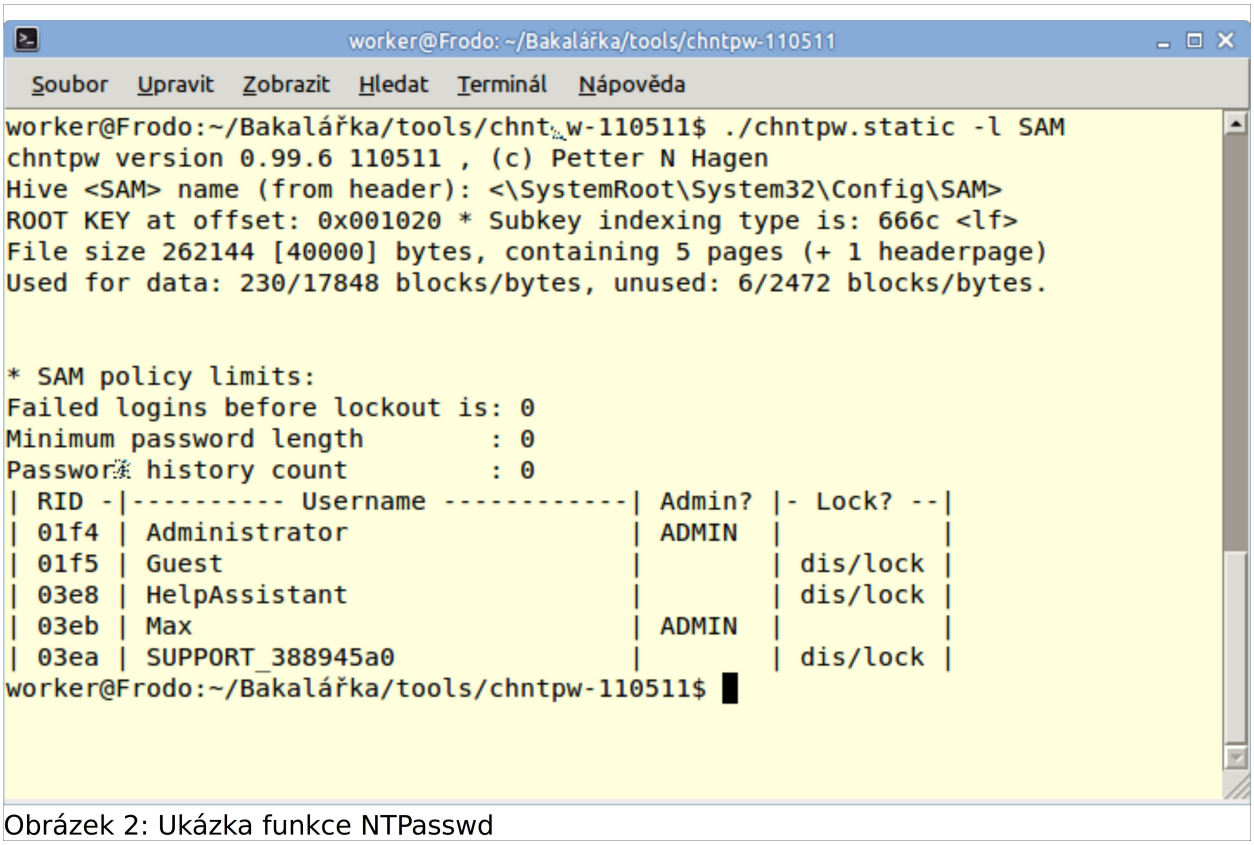
Protože jsou s námi Windows Registry již téměř 15 let, tak se za tu doby objevily nástroje, které se získávání dat z Windows Registrů umožňují a také dokumentují strukturu Windows Registrů. Existuje rozsáhlá studie, která se zabývá pouze dokumentací a rozsahem jednotlivých prací, které se problematikou Windows Registrů zabývají - THE INTERNAL STRUCTURE OF THE WINDOWS REGISTRY [Norris, 2009]

Většina z těchto nástrojů vznikla při forenzním zkoumání Windows Registrů. Hlavním výstupem těchto nástrojů jsou informace jako hesla, registrační klíče, případně se snaží pomocí analýzy zjistit chování uživatele na cílové stanici. Tyto nástroje se také často zabývají “dolováním” dat z paměti, ve které zůstanou otisky Windows Registrů po vypnutí počítače.

Opět se jedná hlavně o zjištění chování a historie uživatele kupříkladu pro kriminalistickou analýzu. Nevýhodou všech těchto nástrojů, je špatná dohledatelnost - dost často se jedná o knihovny, které nejsou udržované, slouží skutečně pouze jedinému účelu a jejich odkazy jsou často nefunkční.

4.1 NTPasswd

Tato ulitita je primárně určená pro restartování administrátorského hesla ve Windows. Její funkce se postupně vylepšovaly a nyní nabízí i základní funkcionalitu pro editaci klíčů a hodnot.



Obrázek 2: Ukázka funkce NTPasswd

Tato utilita je vydávána společně s bootovacím obrazem pro přímé použití jako live distribuce. Je psána v programovacím jazyku C a svůj účel plní velmi dobře. Autorem je Petter Nordahl-Hagen.

4.2 Wine – regedit

Wine je jedním z nejrozsáhlejších projektů, které se snaží reimplementovat Windows API a tím zajistit kompatibilitu aplikací z Windows pro Linux. Tento projekt simuluje

celkovou strukturu Windows – včetně adresářové struktury a proto zde nechybí také simulace Windows Registrů.

Wine v pravém slova smyslu nepoužívá skutečnou strukturu Windows Registrů, pouze se snaží umožnit aplikacím ukládat jejich informace a posléze je tam opět najít. Samotný program „regedit“, se velice podobá skutečnému Windows Regedit32 ale ve skutečnosti pouze čte informace z těchto „simulovaných“ registrů. Neumí otvírat cizí hive a umožňuje pouze exportovat a importovat textové soubory.

Nejedná se tedy o skutečný editor Windows Registrů ale pouze jeho „simulaci“.

4.3 Python-registry

Jako poslední nástroj chci zmínit nástroj „python-registry“. Autorem tohoto projektu je Willi Ballenthin, který se zabývá forenzní analýzou Windows Registrů. On sám napsal modul, která dokáže zobrazovat logickou strukturu Windows Registrů podobně jako to umí nástroj Windows Regedit.

Samotný nástroj je celkem dobře ovladatelný ale bohužel nabízí pouze „read-only“ přístup. Žádné další operace, kromě čtení, nejsou podporovány.

Výhodou tohoto nástroje je dobrá čitelnost zdrojového kodu – který lze nalézt na githubu : https://github.com/williballenthin/python-registry a také samotný jazyk Python, ve kterém je tento nástroj psán.

Jeho velkým omezením je nemožnost jakýchkoliv změn, či jiného grafického rozhraní pro uživatele – otvírání nového souboru, zavření starého, uložení změn atd.

5 Struktura Windows Registrů

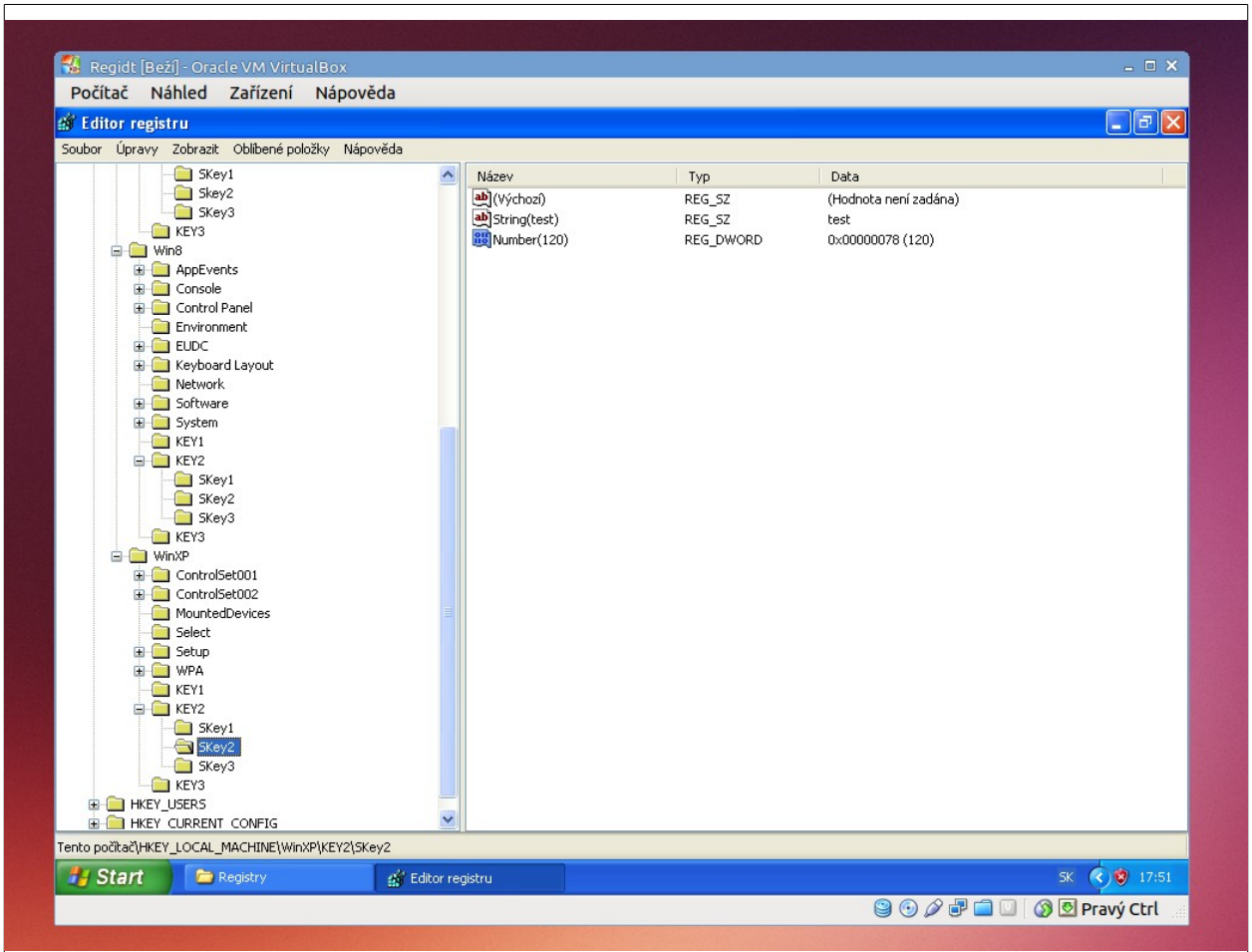
Samotnou strukturu Windows Registrů můžeme rozdělit na logickou a fyzickou. Logickou část tvoří hiearchie klíčů, jejich typ, hodnoty či význam. Fyzikou část chápeme jako samotný způsob zápisu klíčů v binární podobě.

5.1 Logická struktura

Windows Registry jsou navrženy jako hiearchická databáze typu key -> value.
Na samém vrcholu stojí root klíče. Tyto root klíče mají předponu HKEY a jsou reprezentovány pomocí fyzické hive - ale existují i vyjímky.

- HKEY_CLASSES_ROOT
- HKEY_CURRENT_USER
- HKEY_LOCAL_MACHINE
- HKEY_USERS
- HKEY_CURRENT_CONFIG
- HKEY_DYN_DATA
    - Tato hive se tvoří dynamicky a nemá skutečný fyzický soubor - stále existuje ale Windows Xp a vyšší ji již nezobrazují v defaultním zobrazení a není doporučeno jej používat.

Celou strukturu si lze představit i jako jednoduchý souborový systém. Kde klíče reprezentují složky a hodnoty soubory ve složkách. Celkem dobrou představu umožňuje nástroj Regedit, který přímo načítá všechny systémové hive a zobrazuje je jako jednotlivé podklíče.



Obrázek 3: Logická struktura

Každý z klíč má „defaultní“ hodnotu – tato hodnota je označena textem (Default). Každá hodnota klíče musí mít Název, Typ a Hodnotu. Samotný Typ pouze určuje, jak má být hodnota klíče interpretována. V programu Regedit existují tyto typy klíčů, které může uživatel použít:

- REG_SZ = jednoduchý string
- REG_MULTI_SZ = několik stringů oddělených novým řádkem
- REG_DWORD = integer, číselná hodnota
- REG_BINARY = binární hodnota buď jako string a nebo hexadecimální číslo.

5.2 Fyzické umístění souborů

Defaultní umístění Windows Registrů v systému windows je ve složce :

```
%SystemRoot%\System32\Config
```

Jedinou výjímku tvoří soubor s uživatelským nastavením, který je uložen v profilové složce:

```
%SystemRoot%\Profiles\Username
```

Samotné root klíče jsou mapované na těchto lokacích:

* HKEY_LOCAL_MACHINE\SYSTEM:
  - Windows\System32\Config\System
* HKEY_LOCAL_MACHINE\SAM:
  - Windows\System32\Config\Sam
* HKEY_LOCAL_MACHINE\SECURITY:
  - Windows\System32\Config\Security
* HKEY_LOCAL_MACHINE\SOFTWARE:
  - Windows\System32\Config\Software
* HKEY_USERS\.DEFAULT
  - Windows\System32\Config\Default

HKEY_CLASSES_ROOT - je kombinace těchto dvou klíč:
HKEY_LOCAL_MACHINE\Software\Classes a HKEY_CURRENT_USER\Software\Classes

Existuje taktéž výjimka, při takzvané „virtualizaci“ registrů. Tato funkce přišla s Windows Vista, kdy se zamezilo aplikacím zapisovat do některých částí Windows Registrů. [MSDN, 2013]

Toto vyvolalo kolaps se staršími aplikaci a proto bylo nutné vytvořit virtualizaci, která je umístěna v uživatelově složce a pokud se aplikace pokusí přečíst zakázanou oblast, aplikace je přesměrována do uživatelova lokálního úložiště.

5.3 Druhy souborů

Rozlišujeme několik druhů fyzických souborů Windows Registrů, které se liší podle jejich přípony a každý z těchto souborů má jiný účel.

5.3.1 Hive Files

Tyto soubory obsahují skutečný obsah Windows Registrů. Nemají žádnou příponu.

5.3.2 LOG Files

V těchto souborech se evidují veškeré změny, které systém Windows provádí s danou hive. Pokud se poškodí nebo ztratí, lze z tohoto LOG souboru znovu vrátit korespondující hive do původního stavu. Před uložením jakýchkoliv dat do skutečné hive, se změny ukládájí do tohoto souboru. Po úspěšném uložení, se tyto změny zapíší i do skutečného hive souboru. [Norris, 2009] Každý soubor má příponu .LOG

5.3.3 SAV Files

Tyto soubory obsahují přesnou kopii hive souborů. Vytvářejí se při instalaci Windows a to ještě předtím, než se přejde do grafického režimu, který by mohl způsobit pád celého systému. Tyto soubory složí k obnově systému, například při spuštění “opravy” z instalačního CD. Každý soubor má příponu .SAV

5.3.4 ALT Files

Alternativní kopie celé hive. Vytváří se pouze u důležitých systémových klíčů. Každý soubor má příponu .ALT - *tyto soubory už nejsou více používány od Windows XP.* [Kokoreva, 2004]

5.4 Fyzická struktura

5.4.1 Hive

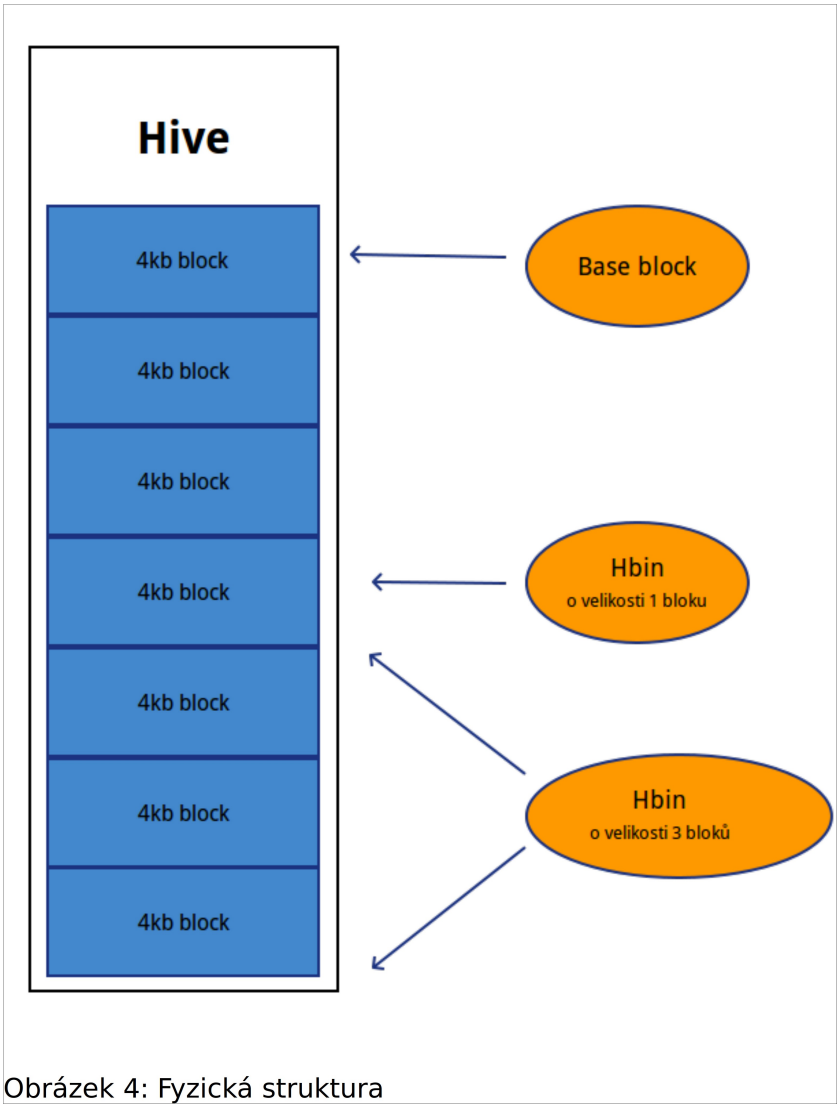
Hive je fyzický soubor, ve kterém je uložen obsah databáze Windows Registrů. [Russinovich, 1999] Hive je tvořena z určitého množství bloků. Při zvětšení nebo zmenšení je vždy zvětšena přesně o násobek velikosti tohoto bloku. Proto vždy platí :

```
velikost_hive / velikost_bloku = počet_bloků;
```

Každý blok má velikost 4Kb (4096 bytů). První blok na začátku hive se nazývá „base block“, tento blok má v sobě uložena důležitá data o konkrétní hive.

První 4byty slouží pro určení typu souboru a mají hodnotu „regf“. Tato hodnota idetifikuje že se jedná o soubor s databází Windows Registry. Mezi další důležité informace patří poslední datum zápisu, verze Windows Registru, kontrolní součet a cesta k souboru. Dále také obsahuje odkaz na první a poslední hbin v hive pomocí offsetu. [Morgan, 2009]

Hive se skládá za hbinů, každý bin musí být opět dělitelný velikostí bloku.



Obrázek 4: Fyzická struktura

5.4.2 HBin

Hbin je struktura, která obsahuje jednu nebo více buněk. Na začátku každého hbinu je jeho hlavička. Tato hlavička je dlouhá 32bytů a stejně jako hive na svém začátku obsahuje 4byty pro určení typu, tyto byty mají hodnotu „hbin“. V hlavičce se taktéž nachází odkaz na první hbin celé hive, vlastní velikost a realtivní odkaz na předchozí hbin.

5.4.3 Buňky

Buňky jsou nejmenší části hive. Vyplňují jednotlivé hbiny a mohou obsahovat buď klíč a nebo hodnotu. Jejich hodnota není pevně určena, mohou se roztahovat a jedna buňka může vyplnit celý hbin. Není možné aby buňka stála samostatně, vždy musí patřit do některého z hbinů.

Pro další informace o fyzické struktuře doporučuji přečíst - The Windows NT * Registry File Format, Timothy D. Morgan.

6 Návrh aplikace

Samotná aplikace by měla sloužit k tomu, aby měli uživatelé komfortní grafický editor, na který jsou navyknutí z Windows. Není snahou této práce aby byl výsledek naprosto stejný ale aby bylo možno bez problémů k Windows Registrům přistupovat a nebylo nutné používat proprietární API Microsoftu.

6.1 Požadavky

Při návrhu aplikace byl brán zřetel na to, aby tato aplikace splňovala tyto požadavky:

- Aplikace musí být schopná otevřít jakýkoliv soubor s daty Windows Registrů.
- Aplikace oznámí uživateli, pokud je daný soubor poškozen a nebo to není validní soubor Windows Registrů.
- Aplikace bude přehledně zobrazovat logickou strukturu Windows Registrů
- Aplikace umožní pomocí stromové struktury procházení jednotlivých klíčů a jejich hiearchie
- Aplikace bude schopná vypsat hodnoty jednotlivých klíčů
- Aplikace bude umožňovat přidání či odebrání klíče do stromu
- Aplikace bude umožňovat přidání, odebrání či editaci hodnoty klíče.

- Aplikace by měla splňovat základní prvky objektového modelu a uživatelské přístupnosti.

6.2 Použité technologie

6.2.1 Python

Jako hlavní programovací jazyk byl vybrán Python, konkrétně verze 2.7.5. Vyšší verze pythonu >= 3 pracují s jinou syntaxí, která již není zpětně kompatibilní.

Mezi hlavní přednosti programovacího jazyka Python patří jednoduchost jeho syntaxe, rozsáhlost dostupných knihoven, dokumentace funkcí, multiplatformní podpora a další.
Tento jazyk byl také vybrán z důvodu možnosti na napojení na knihovnu hivex, která tvoří jádro aplikace a poskytuje binding pro ovládání z Pythonu. Pro vytvoření grafického jádra byl použit toolkit wxWidgets.

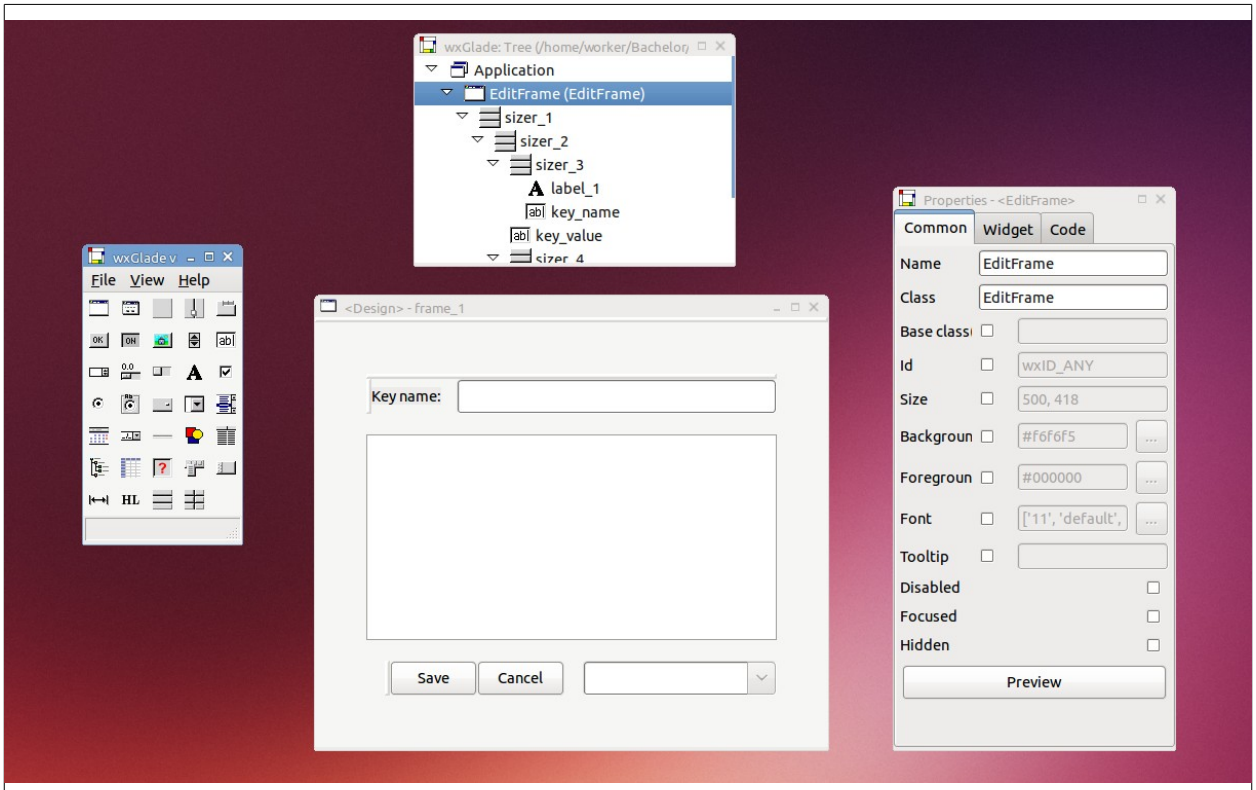
6.2.2 wxWidgets

Jedná se o komplexní sadu grafických widgetů, který pomáhají programátorovi při návrhu grafického prostředí – jako jsou dialogy, tlačítka, rámy a další komponenty. Tato sada widgetů je také multiplatformní, podporuje Windows, Mac a i různá grafická prostředí v Linuxu – GTK, X11. Další výhodou této knihovny je nástroj WxGlade.

6.2.3 WxGlade

WxGlade je program navržený pro design prostředí knihovny wxWidgets. Tento program umožňuje pomocí metody drag and drop vytvořit GUI struktura formuláře a implementovat různé widgety bez nutnosti psaní kódu. Samotný kód lze potom generovat do souboru a využít ho pro jednotlivé komponenty.

Program taktéž nabízí velké možnosti nastavování jednotlivých komponent a propojování události s vnitřními funkcemi.



Obrázek 5: Náhled programu WxGlade

6.2.4 Hivex

libhivex - Windows Registry "hive" extraction library

Tato knihovna je určena pro přímou – fyzickou – změnu v souborech Microsoft Registrů. Oproti jiným alternativám tato knihovna skutečně mění přímo fyzickou (binární) strukturu v souborech a nepoužívá žádné cizí API či textovou interpretaci.

Knihovna hivex je součástí většího balíčku knihoven zvaného libguestfs (http://libguestfs.org), který slouží jako větší sada nástrojů pro manipulaci s virtuálními obrazy disků.

Mezi veliké přednosti této knihovny patří její propracovanost a velké množství testů, které potvrzují, že tato knihovna ukládá a čte hodnoty správně. Důraz je taktéž kladen na stabilitu. Hivex razí teorii lepší než něco pokazit je na to nesahat. Tato teorie je, vzhledem k povaze a důležitosti záznamů ve Windows Registrech, skutečně správná.

Dalším důvodem proč vybrat tuto knihovnu bylo napojení na python, které snadno umožňuje přistupovat k funkcím knihovny. Samotná knihovna je napsána v jazyce C a krom pythonu podporuje také Ruby, Ocaml a Perl. Poslední výhodou této knihovny je její dokumentace, která velice dobře mapuje veškeré potřebné funkce a možnosti s kterými se lze setkat při programování.

Autorem této knihovny je Richard W.M. Jones, z firmy Redhat.

7 Implementace

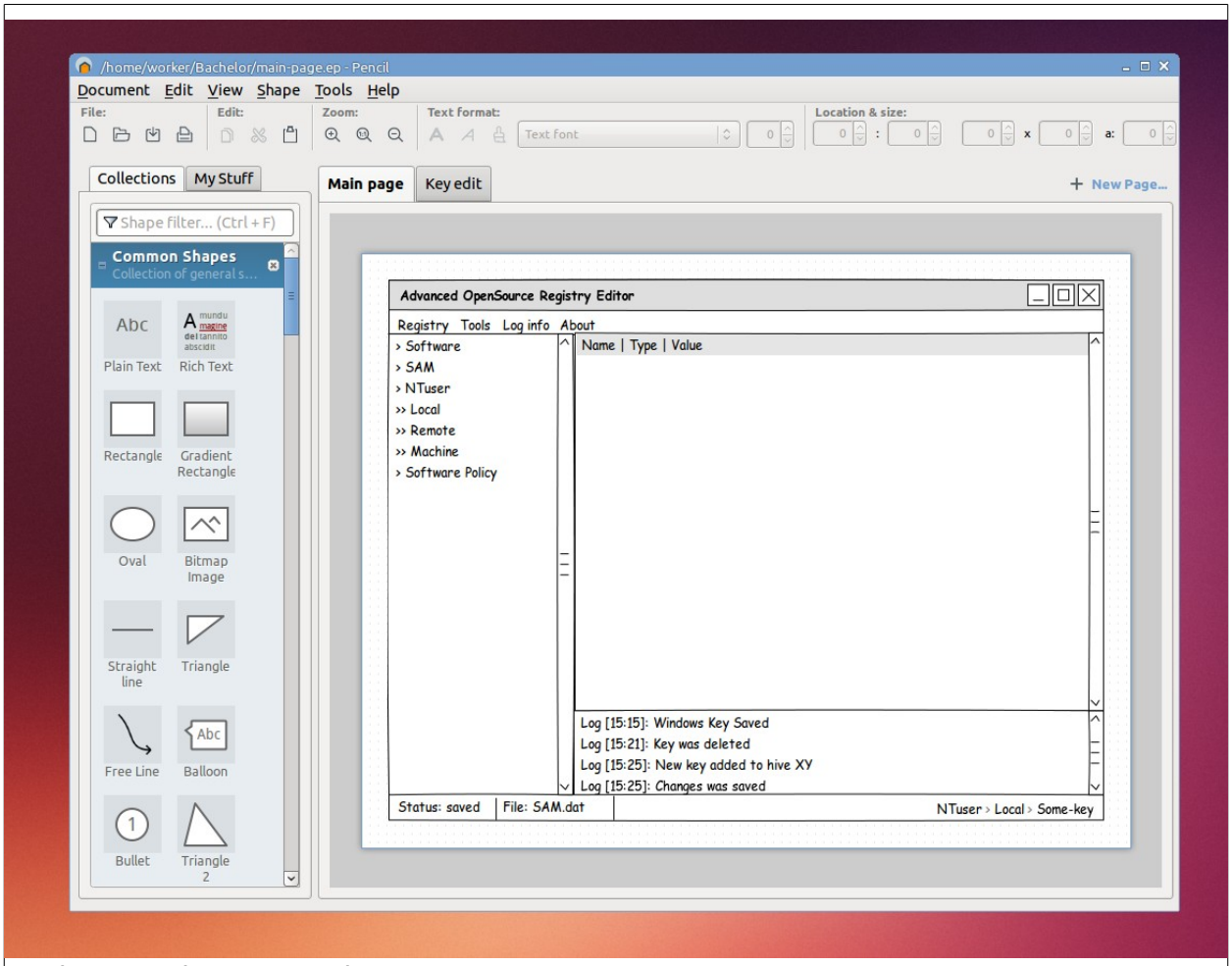
7.1 Návrh grafického rozhraní

Grafické rozhraní je velmi důležité. Zvláště když uživatel potřebuje pracovat s více daty, která jsou hierarchicky strukturována, je nutné vidět kde se nachází, jaký typ mají tyto data a také jaké hodnoty. Všechny tyto údaje by měli být vidět na první pohled a nemělo být moc obtížné se k nim dostat.

Při návrhu aplikace PyRegedit byl brán také ohled na to, že uživatelé patrně znají oficiální program Windows Regedit, který je zabudován v systému Windows a proto by se struktura neměla drasticky lišit. Tedy uživatel Regeditu by měl být schopen bez větších potíží použít i PyRegedit. Některé vlastnosti jsou samozřejmý jiné ale pouze tam, kde to bylo z důvodu technických omezeních a nebo z důvodu většího komfortu.

7.1.1 Pencil

Pro skicy (wireframe) možného vzhledu grafického rozhraní byl použit program Pencil. Pencil je open-source scatching program, který disponuje širokou knihovnou tvarů a ovládacích prvků. Je tu možné pomocí metody „drag and drop“ vytvářet tlačítka, rámy, text, políčka a další. Díky napojení na online knihovnu clipart, je možné vyhledávat z programu přímo v celé databázi a výsledek rovnou použít. Zde můžete vidět ukázku návrhu pomocí Pencilu - Obrázek 6: Návrh rozhraní v programu Pencil

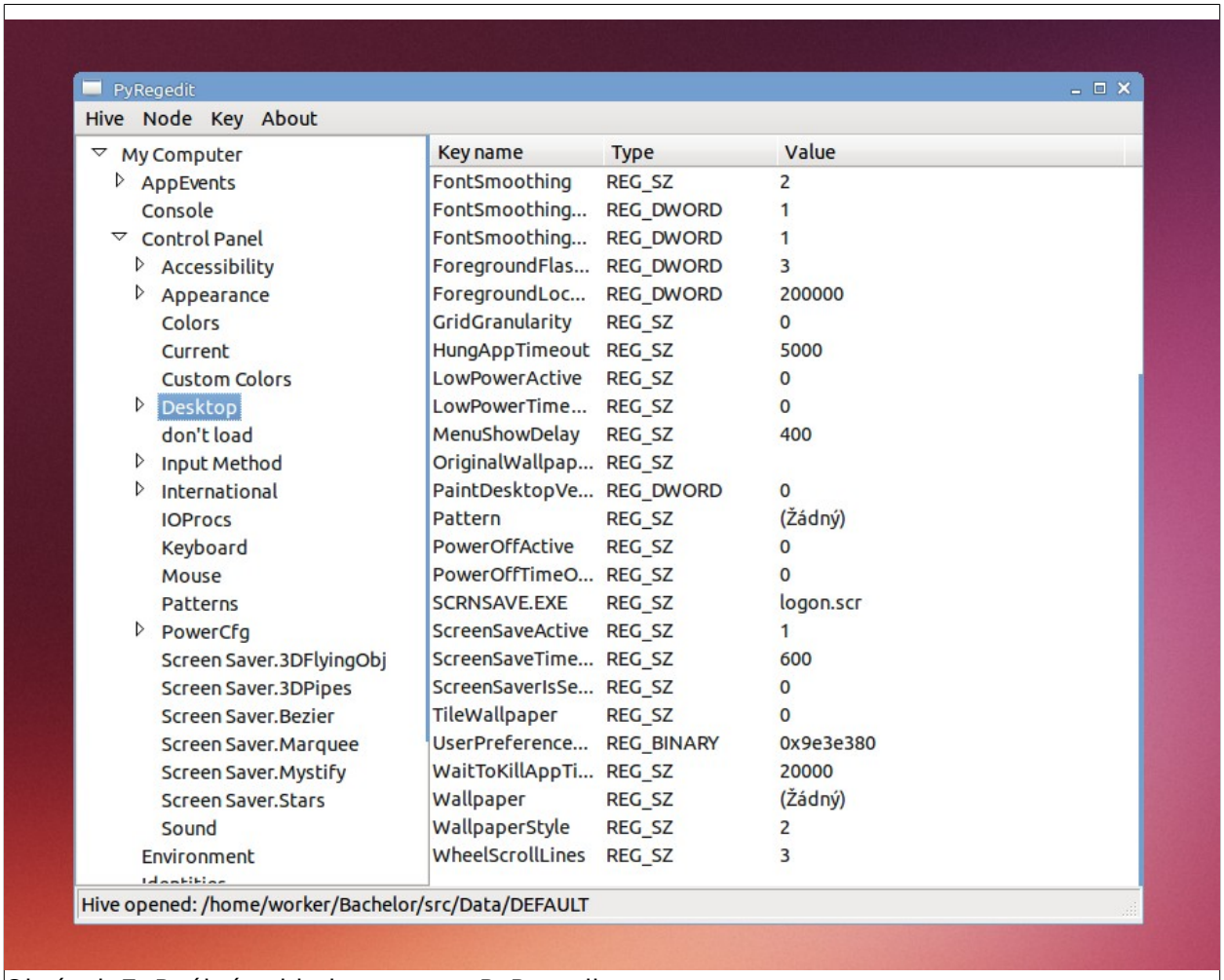


Obrázek 6: Návrh rozhraní v programu Pencil

Tento program taktéž umožňuje export výsledku do různých formátů – PNG, PDF s různými nastaveními – popisem stránky, odsazením. Za projektem Pencil stojí vývojáři z firmy Mozilla.

7.1.2 Prvky grafického rozhraní

Na obrázku: Obrázek 7: Reálný vzhled programu PyRegedit můžete vidět jak vypadá skutečný program. Tento program se v zásadě skládá ze dvou hlavních rámů. První rám je slouží pro prohlížení dat (MainFrame) a druhý pro manipulaci s daty (EditFrame).



Obrázek 7: Reálný vzhled programu PyRegedit

7.1.3 MainFrame

Tento rám slouží jako hlavní část programu. Při prvním spuštěním je defaultně viditelný a jeho okno má velikost 800x600px, což by mělo vyhovovat i starším rozlišením s 1024x768px.

Toto hlavní okno má všechny základní systémové prvky jako tlačítko pro zavření, maximalizaci, minimalizaci. V hlavičce tohoto rámu vidět název programu – PyRegedit. Pokud uživatel provádí některé změny s hive a tyto změny je potřeba následně uložit, je o tom informován v hlavičce pomocí textu - „MODIFIED“, po řádném uložení je tento text změněn na „SAVED“.

Samotný rám obsahuje několik oddělených komponent, jsou to :

- MenuBar
- Statusbar
- TreeView
- Listview

7.1.4 MenuBar

Menu bar slouží pro přístup k základním operacím, které uživatel může provádět. Existují tu tři SubMenu:

- Hive
  - Toto volba dovoluje uživateli manipulovat s hive:
    - otevřít novou hive (open)
    - uložit změny do hive (save)
    - aktualizovat hive (reload)
    - zavřít hive (close)
- Node
  - Toto volba dovoluje uživateli měnit top klíče (nody):
    - přidat nový top level klíč (add node)
    - smazat top level klíč a jeho potomky (remove node)
- Key
  - Tato volba dovoluje uživateli měnit hodnoty klíče:
    - přidat nový klíč (add new key)
    - odebrat klíč (remove key)
- About
  - Tato volba je pro informace o aplikaci, uživatel zde nalezne krátký popis, verzi a odkaz na git hub repository.

7.1.4.1 StatusBar

Status bar slouží jako zjednodušený log. Uživatel tu může nalézt hlášky o vykonávané operaci a taktéž tu vidí, který soubor má otevřen, zda se operace povedla, případně jiné užitečné informace.

7.1.4.2 TreeView

TreeView slouží pro zobrazení stromové struktury hlavní (top) klíčů celé hive. Jako Root zde slouží „imaginární" klíč, s názvem „My Computer". Tento klíč skutečně neexistuje a pouze reprezentuje strukturu, která je často známá ze zobrazení v Regedit, kde je taktéž root klíč zobrazen.

Při prvním spuštění jsou vypsány pouze klíče do 1. úrovně. Každý klíč, který má podklíče je označen pomocí šipky a po kliknutí na něj lze vypsat 2. úroveň. Tímto způsobem lze efektivně procházet celou strukturu hive.

Pokud klíč obsahuje některé hodnoty, po jeho označení se automaticky aktualizace výpis hodnot v ListView.

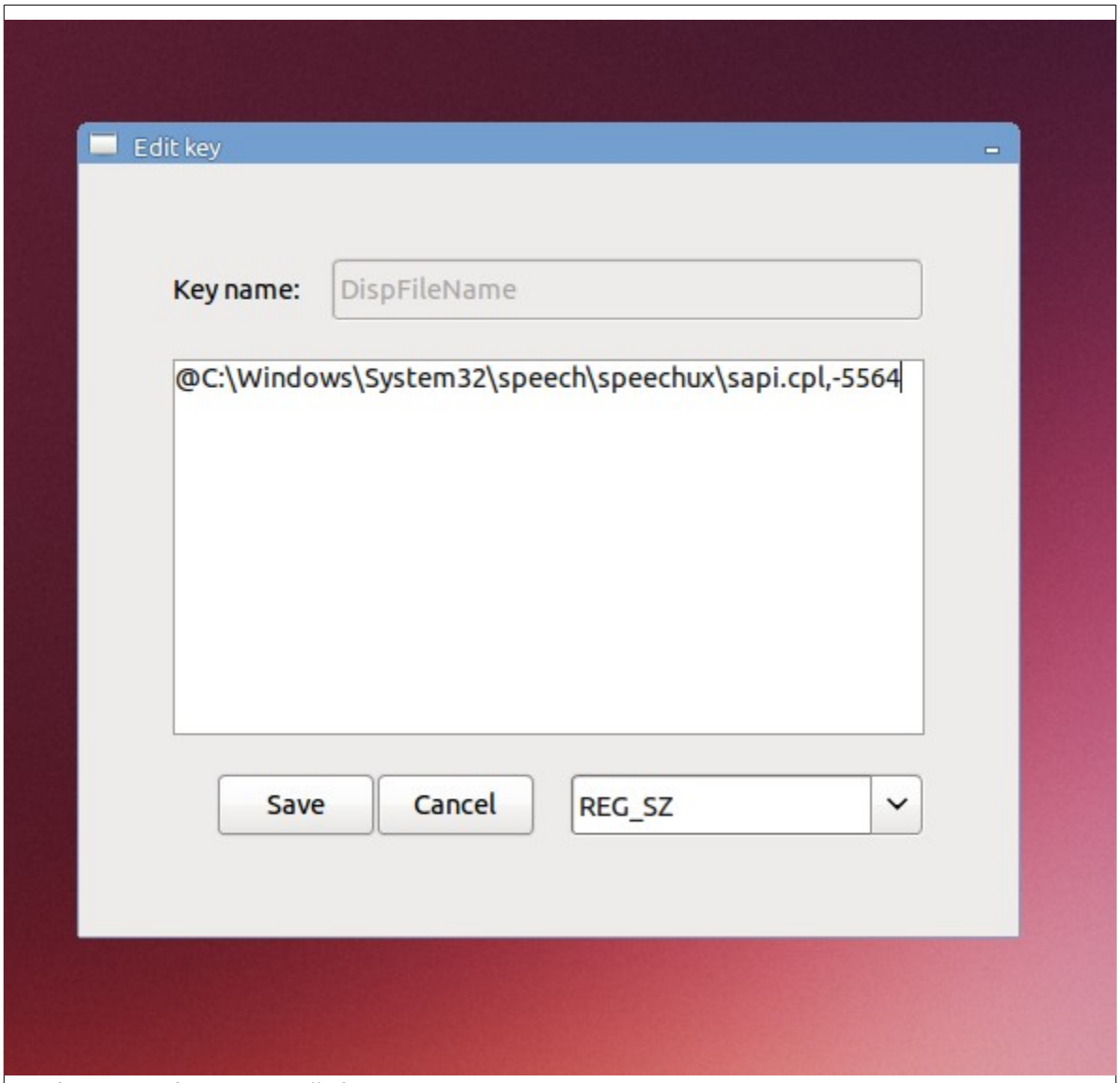
7.1.4.3 ListView

ListView slouží pro přehledné zobrazení hodnot určeného klíče. Jeho zobrazení obsahuje 3 nedůležitější sloupce – Název, Typ a Hodnotu. Zobrazení ListView je aktualizováno po každém vybrání klíče či změně hodnoty. Klíče jsou seřazeny podle abecedního řazení – dle názvu. Každý klíč lze editovat pomocí dvojkliknutí, po tomto dvojkliknutí je zobrazen editační formulář.

7.1.4.4 EditFrame

Tento rám slouží pro editaci či přidání klíče. Obsahuje pouze základní komponenty, které umožňují změnit hodnotu, jméno či typ daného klíče. Pomocí dvou tlačítek lze změny uložit či stornovat.

Rozdíl mezi editačním oknem a oknem pro přidání nového klíče je v zakázání změny políčka pro název klíče. Z technických důvodů není možné změnit název již funkčního klíče. Stejně tak editační formulář má načteny všechny základní hodnoty a přednastavený typ, který pochází z klíče, který edituje.



Obrázek 8: Náhled editačního dialogu PyRegedit

7.2 Tvorba programu

Z důvodu oddělení architektury a lepší škálovatelnosti byl pro návrh aplikace použit model **MVC**. Díky tomuto oddělení je mnohem snažší přidávat další grafické komponenty, bez zásahu do logiky již funkčních komponent. Stejně tak samotné oddělení jednotlivých tříd slouží pro lepší přehled logiky celého návrhu.

7.2.1 Implementace MVC

Návrhový vzor MVC se skládá ze 3 základních komponent, jedná se o

- Modely
- View
- Controllery

Při implementaci tohoto návrhového vzoru jsem se držel těchto pojmenování a proto i adresářová struktura celé aplikace reflektuje tento návrh. Tedy Modely nalezneme ve složce Models, View ve složce View a Controllery ve složce Controllers.

7.2.1.1 Views

Views se dá česky přeložit jako zobrazení a slouží pro grafickou prezentaci dat uživateli. V aplikace PyRegedit jsou pod tímto míněny všechny grafické prvky, které tato aplikace potřebuje. Patří sem tedy MainFrame, Dialogová okna, komponenty jednotlivých rámů → menuBar, statusBar atd.

Každý z Views většinou dědí z ekvivalentního widgetu WX. Samotný obsah třídy je inicializace widgetu a nastavení jeho vlastního layoutu a obsahu.

Skutečný obsah složky Views vypadá takto:

- __init__.py
  - inicializační soubor pro python
- Dialog.py
  - soubor ve kterém jsou uloženy všechny dialogy které aplikace používá
    - AboutDialog
    - AddNodeDialog
- Frame.py
  - soubor obsahuje definice pro hlavní rámy celé aplikace, z důvodu složitosti je tento kód většinou tvořen generováním z programu WxGlade
    - MainFrame
    - EditFrame
- MenuBar.py
  - definice menu celé aplikace

- jsou zde uloženy ID, které mají jednotlivé položky

7.2.1.2 Models

Modely mají sloužit pro samotnou logiku aplikace a práci s daty. V případě této aplikace jsou implementovány tyto 3 modely.

- HivexManager.py
  - Tento model slouží pro komunikaci s knihovnou hivex.
  - Stará se o základní funkce, jako je například vybrání klíčů pro zobrazení ve stromové struktuře.
  - Tento model taktéž obsluhuje většinu změň v interpretaci hodnot
    - klíč do čitelné hodnoty
    - čitelná hodnota zpět do formátu pro uložení klíče
- File.py
  - Tento model je jednoduchá abstrakce pro manipulaci se souborem.
  - Stará se o otevření souboru a o ověření, že soubor existuje a lze s ním pracovat jako s hive
    - Soubor musí mít první 4b unikátní řetěz „regf“.
- Type.py
  - Zde se v pravém slova smyslu nejedná o model.
  - Z důvodu absence typu Enum v pythonu, se tento model stává jeho náhradou. Jsou zde uloženy všechny typy klíčů a taktéž tu je jejich textová interpretace.

7.2.2 Controllers

Controller neboli řadič. Měl by sloužit k tomu, aby kontroloval činnost Modelů a předával správný View pro uživatele. V implementaci aplikace PyRegedit se jedná o jediný soubor, který zastává veškerou aplikační logiku.

- Controller.py
  - Tento soubor inicializuje celý aplikační proces.
  - Obsluhuje uživatelské události – kliknutí na tlačítko, zavření boxu atd.
  - Kontroluje validitu dat – pokud soubor neexistuje sdělí to uživateli
  - Spadá pod něj veškerá logika ohledně řízení přístupu k hive – ukládání, mazání, zavírání.

7.2.3 Struktura aplikace

Jak už jsem popsal v předchozích kapitolách, struktura celé aplikace je podřízena modelu MVC. Skutečná struktura aplikace vypadá tedy takto:

- Models
  - __init__.py
  - File.py
  - HivexManager.py
  - Type.py
- Controllers
  - __init__.py
  - Controller.py
- Views
  - __init__.py
  - Dialog.py
  - Frame.py
  - MenuBar.py
- Data
- app2.py
- Readme.md
- registry-icon.png

8 Testování

Testování aplikace je jedna z nejdůležitějších částí celého procesu tvorby. Zvláště když se jedná o změny v tak citlivém místě jako jsou Windows Registry. Hlavním účelem celého procesu testů má být zaručení uživateli co největší jistou, že jeho změny budou správně uloženy a interpretovány systém Windows.

8.1 Metodika

Pro průkazné otestování byli vybrány vzorce hive z několika různých verzí systémů. Jednalo se o tyto verze:

- Windows XP
- Windows 7 Professional
- Windows 8.1

U Windows XP byly změny testovány v hive SYSTEM, u Windows 7 a Windows 8.1 u hive NTUSER.DAT

Samotný průběh testování se skládal ze dvou sérií. První série měla prokázat, že data zapsaná pomocí nativní Windows aplikace lze číst pomocí PyRegedit a následně také upravit. Druhá série měla prokázat, že data zapsaná pomocí PyRegedit lze číst a následně upravit pomocí nativní aplikace Windows.

Při každé sérii byli provedeny tyto kroky:

1. Vytvořit ti hlavní klíče (nody)
2. V libovolném klíči vytvořit tři podklíče
3. V libovolném klíči vytvořit dvě různé hodnoty (REG_SZ, REG_DWORD)
4. Přečíst všechna data a ověřit správnost.
5. Změnit hodnotu dvou hodnot.
6. Smazat jeden klíč
7. Přečíst všechna data a ověřit správnost.

Ověřování správnosti změn se vždy provádělo v opačném systému. Pro čtení hodnot ve Windows byl použit program Windows Regedit, který byl virtualizován pomocí Virtual Boxu.

Poznámka: Regedit umožňuje připojit nový podregister ale pouze v takovém registru, kde má uživatel právo zapisovat – jinak je tato možnost zakázána ale program vám to nijak nesdělí. Pokud chcete připojit nový register, je nutné mít označen podklíč HKEY_LOCAL_MACHINE

9 Výsledky

9.1 Windows → Linux

Výsledky testování správnosti zápisu z Windows do Linuxu.

Testované operace	**Testované verze hive**		
	Windows XP	Windows 7	Windows 8.1
Vytvořit tři hlavní klíče ve Windows	SPRÁVNĚ	SPRÁVNĚ	SPRÁVNĚ
Vytvořit tři podklíče v hlavním klíči ve Windows	SPRÁVNĚ	SPRÁVNĚ	SPRÁVNĚ
Vytvořit dvě nové hodnoty v libovolném klíči (REG_SZ, REG_DWORD) ve Windows	SPRÁVNĚ	SPRÁVNĚ	SPRÁVNĚ
Změnit dvě hodnoty na Linuxu a ověřit výsledek ve Windows.	SPRÁVNĚ	SPRÁVNĚ	SPRÁVNĚ
Smazat jeden klíč v Linuxu a ověřit výsledek ve Windows.	SPRÁVNĚ	SPRÁVNĚ	SPRÁVNĚ

V této sérii byl jako výchozí program použit Windows Regedit. V tomto výchozím programu byli provedeny příslušné operace a tyto operace byli posléze ověřeny na opačné platformě v nástroji PyRegedit. Všechny změny byli stále prováděny na stejném fyzickém souboru.

Všechny testované operace proběhly bez problému.

9.2 Linux → Windows

Výsledky testování správnosti zápisu z Linuxu do Windows.

Testované operace	**Testované verze hive**		
	Windows XP	Windows 7	Windows 8.1
Vytvořit tři hlavní klíče v Linuxu	SPRÁVNĚ	SPRÁVNĚ	SPRÁVNĚ
Vytvořit tři podklíče v hlavním klíči v Linuxu	SPRÁVNĚ	SPRÁVNĚ	SPRÁVNĚ
Vytvořit dvě nové hodnoty v libovolném klíči (REG_SZ, REG_DWORD) v Linuxu	SPRÁVNĚ	SPRÁVNĚ	SPRÁVNĚ
Změnit dvě hodnoty na Windows a ověřit výsledek v Linuxu.	SPRÁVNĚ	SPRÁVNĚ	SPRÁVNĚ
Smazat jeden klíč ve Windows a ověřit výsledek v Linuxu.	SPRÁVNĚ	SPRÁVNĚ	SPRÁVNĚ

V opačné serii byl jako výchozí systém Linux a aplikace PyRegedit. Všechny operace byli prováděny opačně a výsledek byl ověřován ve Windows v aplikaci Regedit.

Všechny testované operace proběhly bez problému.

10 Distribuce aplikace

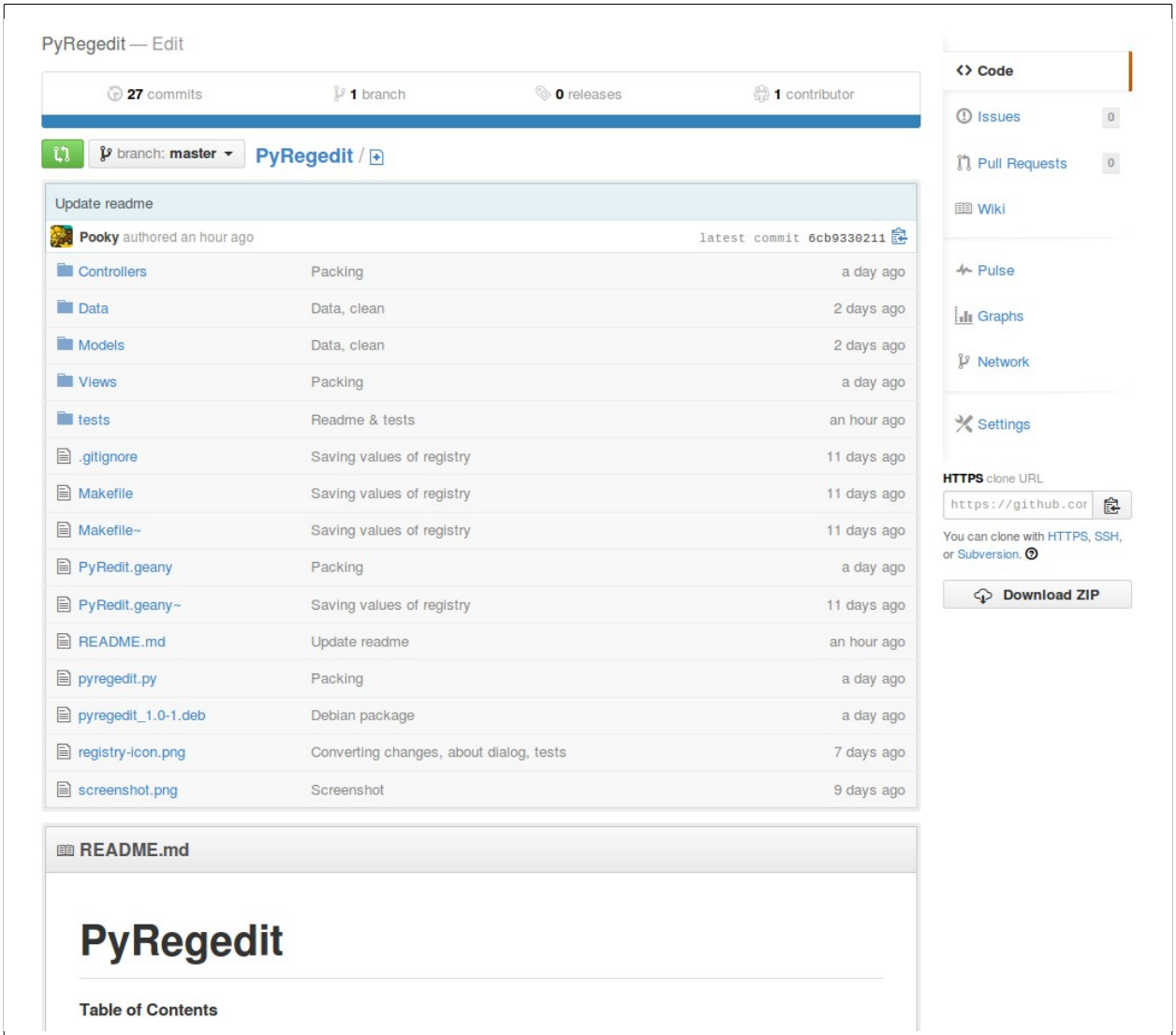
10.1 GitHub

Program PyRegedit je volně dostupný na úložišti GitHub. GitHub je moderní verze systému na správu verzí projektu. Jeho předchůdcem byl systém SVN či Merculiam. GitHub se vyznačuje hlavně svými komunitními prvky a snadným přístupem ke zdrojovým kódům a možností je měnit.

Samotný program lze najít v repositáři se všemi potřebnými soubory k jeho spuštění a taktéž s náležitou dokumentaci v souboru README.md

Společně s tím je program zabalen jako debian balíček, který může uživatel nainstalovat včetně všech jeho závislostí.

GitHub repozitář se nalézá pod touto url : https://github.com/Pooky/PyRegedit



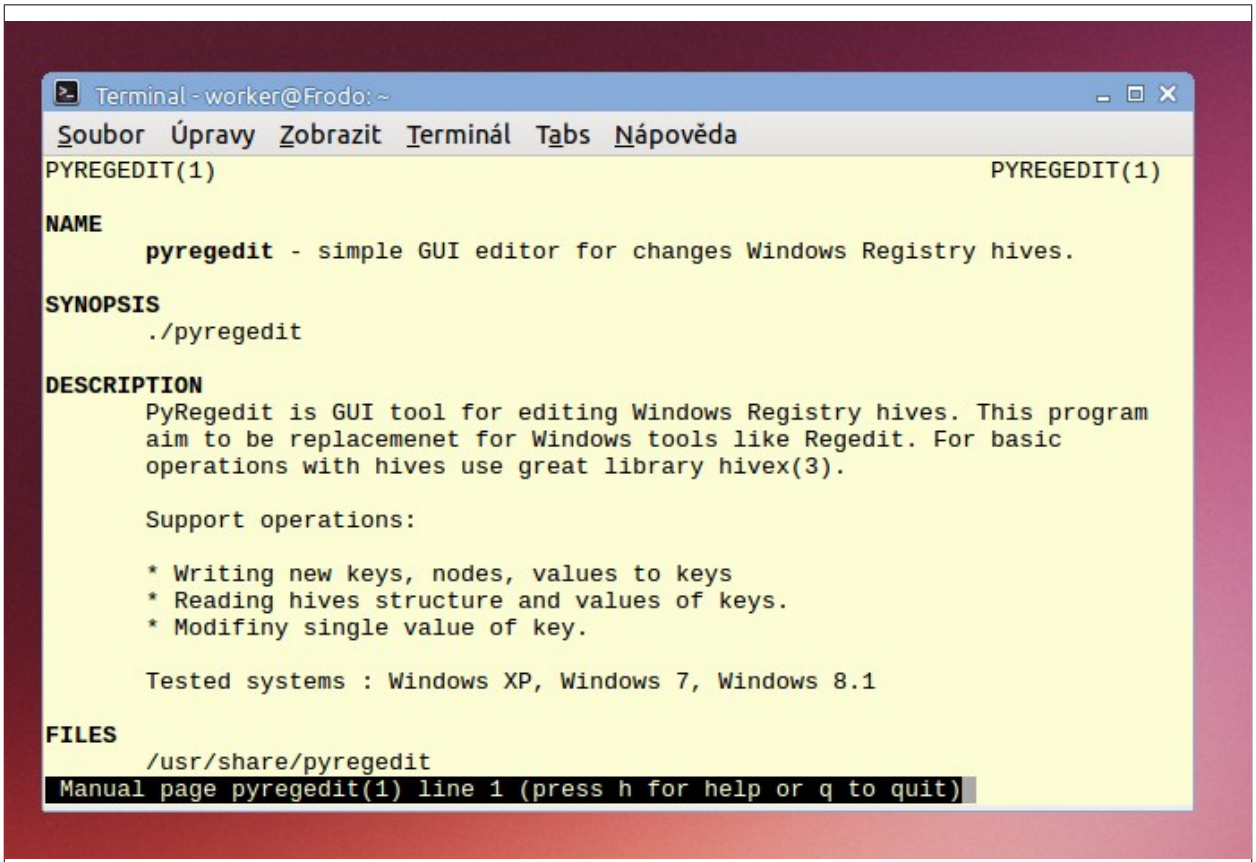
Obrázek 9: GitHub repozitář s aplikací

10.2 Manuálové stránky

Kromě dokumentace, která se naléza na githubu, jsou pro uživatele taktéž dostupné manuálové stránky. Tyto manuálové stránky lze volat pomocí příkazu :

```
man pyregedit
```

Manuálové stránky popisují základní funkce programu. Stejně tak zde uživatel může najít jednotlivé cesty, kde je program instalován. Jako další část manuálových stránek je popis fyzických cest, na kterých se nachází HKEY klíče ve Windows. V poslední části tu je tu také uveden ukázkový příklad.



Obrázek 10: Manuálové stránky PyRegedit

10.3 Instalace

Instalaci můžeme rozdělit na dvě možnosti. Uživatel může využít možnost a instalovat již připravený debian balíček a nebo si stáhnout zdrojový kod a spustit aplikaci přímo pomocí pythonu.

Pro instalaci a správný běh aplikace jsou potřeba tyto balíčky:

- Python 2.7 – balíček python2.7
- Hivex >= 1.3 – balíček python-hivex a libhivex0
- WxPython – balíček python-wxgtk2.8

10.4 Instalace z debian balíčku

Debian balíček se nalézá v github repozitáři:

https://github.com/Pooky/PyRegedit/raw/master/pyregedit_1.0-1.deb

Po jeho stažení ho můžeme instalovat pomocí tohoto postupu:

1. Instalujeme gdebi - sudo apt-get update && sudo apt-get install gdebi
2. Nainstalujeme balíček - sudo gdebi pyregedit_1.0-1.deb
3. Můžeme si prohlédnout manuálové stránky – man pyregedit
4. Spustíme finální program – pyregedit

Samotný debian balíček lze instalovat také alternativní cestou, pomocí dpkg

```
dpkg -i pyregedit_1.0-1.deb
```

Tento balíček může být také v budoucnu distribuován pomocí privátního repozitáře – PPA.

10.5 Použití aplikace

Při použití aplikace je brát zřetel na odlišnosti fyzické a logické struktury. V předchozích kapitolách byly tyto rozdíly popsány. Pokud chceme aplikaci správně používat musíme mít základní informace o Windows Registrech.

Musíme vědět, jaký klíč chceme změnit a v jakém fyzickém souboru se tento klíč nalézá. Fyzické umístnění souborů je popsáno v kapitole 5.2 Fyzické umístění souborů a taktéž musíme správně připojit disk, na kterém máme uložený soubor s Windows soubory.

10.6 Ukázkový příklad

Pro lepší pochopení, si uvedeme jednoduchý příklad. Dejme tomu, že chceme změnit název firmy, kterou jsme uvedli při instalaci Windows. A použijeme k tomu program PyRegedit.

Víme že potřebujeme změnit hodnotu klíče „RegisteredOrganization“, který se dle manuálu nalézá na této lokaci:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion
```

Z kapitoly 5.5 víme, že HKEY_LOCAL_MACHINE je root klíč, který je pouze „logický“, tedy nikoliv skutečně fyzický. Protože potřebujeme HKEY_LOCAL_MACHINE\SOFTWARE tak víme, že tento klíč už fyzický soubor má – a nalézá se na umístění:

```
Windows\System32\Config\Software
```

Předpokládejme, že máme namapovaný disk s Windows pod „/mnt/disk_windows/“ a můžeme do tohoto oddílu číst a zapisovat.

Postup bude následující:

- Otevřeme program PyRegedit - „pyregedit“
- Z horní nabídky vybereme „Hive“ → „Open“.
- Vybereme soubor, v našem případě je to :
  - „/mnt/disk_windows/windows/System32/Config/Software“
- Po otevření souboru v levém panelu najdeme cestu :
  - Microsoft → Windows NT → CurrentVersion
- V pravém panelu vybereme daný klíč „RegisteredOrganization“
- Změníme hodnot a klíč uložíme.
- Uložíme změny do celé hive pomocí „Hive“ → „Save“

Upozornění:

Aplikace požaduje přístup k soboru a také možnost zápisu, do složky kde je soubor umístěn – vytváří se tam backup soubor. Pokud připojujete oddíl s windows, je potřeba zkontrolovat práva, případně umask.

11 Budoucnost

Aplikace PyRegedit není určitě kompletní. Zatím je v prvotní verzi 1.0 a existuje mnoho dalších možností rozšíření, které ale přesahují moje časové možnosti a taktéž rozsah této práce.

Tato aplikace by si rozhodně zasloužila dopracovat část, která by více usnadňovala manipulaci s oddílem Windows. V současné době tato aplikace počítá pouze se souborem, který bude validním souborem Windows Registrů.

Přesto v praxi je potřeba mít namapován soubor s Windows – který převážně využívá souborový systém NTFS. S tím je spojeno mnoho problémů. Oddíl musí být správně namapován, musí tu probíhat error checking, který ochrání hive když se tento oddíl znenadání odpojí a taktéž práva pro zápis, čtení, správný název odílu a mnohé další.

Taktéž by bylo možné dopracovat jednoduchý setup, který by uživatele provedl a našel by za něj z správného oddílu validní hive a pomohl mu při hledání správné cesty, která mapuje logickou strukturu do té fyzické.

12 Závěr

Díky programu PyRegedit by nyní mělo být snažší přistupovat k souborům Windows Registrů a provádět změny, které by dříve uživatel musel složitě hledat v manuálech a jednoúčelových utilitách.

Všechny potřebné funkce tohoto programu fungují a jsou otestovány. Tento program umožňuje:

- zobrazit strukturu klíčů
- číst hodnoty klíčů
- přidávat či odebírat klíče
- měnit hodnoty klíčů a mazat je.

Tento program je náležitě zdokumentován, dostupný pro všechny, kteří by se chtěli zapojit do jeho další tvorby. Stejně tak je připraven pro snadnou instalaci do distribucí pomocí debian balíčku.

Cíle práce byly úspěšně splněny.

13 Odkazy

KLÍMA, Martin. Pooky/PyRegedit. [online]. Dostupné z: https://github.com/Pooky/PyRegedit

Security Accounts Manager. [online]. Dostupné z: http://www.beginningtoseethelight.org/ntsecurity/index.htm

Hivex - Windows Registry "hive" extraction library. W.M. JONES, Richard. REDHAT. [online]. Dostupné z: http://libguestfs.org/hivex.3.html

Chapter 24 - Registry Editor and Registry Administration. MICROSOFT. [online]. Dostupné z: http://www.microsoft.com/resources/documentation/windowsnt/4/workstation/reskit/en-us/24_reged.mspx

Why the Windows Registry sucks ... technically. W. M. JONES, Richard. [online]. Dostupné z: https://rwmj.wordpress.com/2010/02/18/why-the-windows-registry-sucks-technically/

WxWidgets: Cross-Platform GUI Library. WXWIDGETS TEAM. [online]. Dostupné z: https://www.wxwidgets.org/

EUROPEAN COMMISSION. Microsoft Case. [online]. 2002. Dostupné z: http://ec.europa.eu/competition/sectors/ICT/microsoft/

14 Seznam použité literatury

HONEYCUTT, Jerry. Microsoft Windows Registry Guide. Microsoft Press, 2005.

KOKOREVA, Olga. Registr Microsoft Windows XP: kompletní průvodce přizpůsobením a optimalizací operačního systému. Vyd. 1. Praha: Computer Press, 2002, xx, 393 s. ISBN 80-722-6783-3.

NORRIS, Peter. THE INTERNAL STRUCTURE OF THE WINDOWS REGISTRY. 2009. Bakalářská práce. Cranfield University.

D. MORGAN, Timothy. The Windows NT Registry File Format. [online]. Dostupné z: http://sentinelchicken.com/research/registry_format/

PROBERT, David. Windows Kernel Internals NT Registry Implementation [online]. Dostupné z: http://wenku.baidu.com/view/5f61780003d8ce2f006623ef

MICROSOFT. Registry Virtualization [online]. 2013. Dostupné z: http://msdn.microsoft.com/en-us/library/windows/desktop/aa965884%28v=vs.85%29.aspx

RUSSINOVICH, Mark. MICROSOFT. Inside the Registry [online]. 1999. Dostupné z: http://technet.microsoft.com/library/cc750583.aspx