

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

JEDNODUCHÉ SDÍLENÍ POMOCÍ SMB PROTOKOLU  
VE SPRÁVCI NAUTILUS

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JAN LIPOVSKÝ

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

# JEDNODUCHÉ SDÍLENÍ POMOCÍ SMB PROTOKOLU VE SPRÁVCÍ NAUTILUS

SIMPLE FILE SHARING VIA SMB PROTOCOL IN NAUTILUS

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JAN LIPOVSKÝ

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. ALEŠ SMRČKA

BRNO 2010

## **Abstrakt**

Bakalářská práce řeší návrh a implementaci doplňku souborového správce Nautilus společně s grafickým uživatelským rozhraním v prostředí GNOME využívající knihovny GTK+ a démonem, který zajišťuje obsluhu příchozích požadavků. Výsledný program bude umožňovat snadné sdílení vybraného adresáře pomocí SMB protokolu.

## **Abstract**

The bachelor's thesis describes implementation of Nautilus Extension with graphic user interface in GNOME desktop environment using GTK+ library. Daemon serves incoming request from GUI and changes configuration file of Samba system. Final program will share selected directory via SMB protocol.

## **Klíčová slova**

Doplňěk správce Nautilus, GNOME Nautilus, GUI, démon, Samba, SMP protokol, D-Bus, PolicyKit

## **Keywords**

Nautilus Extension, GNOME Nautilus, GUI, daemon, Samba, SMB protocol, D-Bus, PolicyKit

## **Citace**

Jan Lipovský: Jednoduché sdílení pomocí SMB protokolu ve správci Nautilus, bakalářská práce, Brno, FIT VUT v Brně, 2010

# Jednoduché sdílení pomocí SMB protokolu ve správci Nautilus

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Aleše Smrčky. Další informace mi poskytl pan Ing. Tomáš Bžatek. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Jan Lipovský  
19. května 2010

## Poděkování

Tímto děkuji vedoucímu práce Ing. Aleši Smrčkovi za pedagogické vedení, podporu a užitečné rady při tvorbě práce. Děkuji také Ing. Tomáši Bžatkovi za odbornou pomoc a cenné rady při řešení bakalářské práce.

©Jan Lipovský, 2010

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

Obsah.....	1
1 Úvod.....	2
2 Základní pojmy a technologie.....	3
2.1 Správce souborů GNOME Nautilus.....	3
2.1.1 Doplněk souborového správce Nautilus .....	4
2.2 Sdílení souborů systémem Samba.....	4
2.2.1 Konfigurační soubor smb.conf.....	5
2.3 Sběrnice D-Bus.....	7
2.4 Systém PolicyKit.....	8
3 Specifikace požadavků.....	9
4 Návrh řešení projektu.....	11
4.1 Doplněk souborového správce Nautilus.....	11
4.2 Grafické uživatelské rozhraní.....	12
4.2.1 Dialogové okno aplikace.....	12
4.2.2 Komunikace přes sběrnici D-Bus.....	12
4.2.3 PolicyKit – kontrola oprávnění.....	13
4.3 Simple File Share démon.....	13
4.3.1 Práce s konfiguračním souborem.....	13
4.3.2 Komunikace přes sběrnici D-Bus.....	14
4.3.3 PolicyKit – kontrola oprávnění.....	15
4.4 Diagram a specifikace případů užití.....	16
5 Implementace projektu.....	19
5.1 Zásuvný modul správce Nautilus.....	19
5.2 Grafické uživatelské rozhraní.....	19
5.2.1 PolicyKit v grafickém uživatelském rozhraní.....	23
5.2.2 Komunikace přes sběrnici D-Bus.....	24
5.3 Implementace Simple File Share démona.....	24
5.3.1 Obsluha příchozích požadavků.....	25
5.3.2 Funkce pracující s konfiguračním souborem smb.conf.....	29
5.3.3 Pomocné funkce.....	29
5.3.4 Funkce implementující obsluhu D-Bus metod.....	34
6 Závěr.....	35
Literatura.....	36
Seznam příloh.....	38

# 1 Úvod

Poslední dobou je pozorován nárůst používání operačních systémů založených na Unixu na stolních počítačích. Mnoho uživatelů přechází z operačního systému MS Windows na jiný operační systém. Nejčastější volbou je jedna z Linuxových distribucí: Fedora, Ubuntu, OpenSUSE, Mandriva, Gentoo, Debian, Archlinux atd. Nejen pro nové, ale i pro současné uživatele operačního systému Linux vznikají aplikace usnadňující nastavení konfiguračních souborů různých programů.

V této práci se budeme zabývat návrhem a implementací právě takového programu, který zpříjemní používání systému Linuxu s desktopovým prostředím GNOME a souborovým správcem Nautilus. Navrhovaná aplikace, pojmenovaná Simple File Share, omezí přímí zásahy uživatele do konfiguračního souboru a usnadní práci při sdílení adresáře pomocí protokolu SMB ve správci Nautilus.

Následující kapitola popisuje aplikace a technologie, které byly použity při návrhu jednotlivých částí projektu. Snaží se objasnit pojmy úzce související se zadaným úkolem a rozebírá základní vlastnosti a nastavení těchto aplikací. Třetí kapitola shrnuje požadavky kladené na projekt jako celek i na jeho konkrétní části. Tyto požadavky jsou brány v potaz při návrhu řešení projektu. Tím se zabývá kapitola čtvrtá, ve které jsou v podkapitolách podrobně popsány a rozebrány návrhy všech částí projektu. Navazující kapitolou je kapitola pátá, která je stejně jako v předchozím případě strukturována do podkapitol obsahujících podrobný popis implementace doplňku souborového správce Nautilus, démona i grafického uživatelského rozhraní. Vyhodnocení práce je popsáno v poslední, tedy šesté, kapitole. V závěru je popsáno praktické využití programu společně s návrhy možného rozšíření.

## 2 Základní pojmy a technologie

V této kapitole jsou popsány základní pojmy a technologie, které byly použity při návrhu a implementaci projektu. Konkrétně se zabývá vysvětlením problematiky týkající se aplikace Nautilus, sběrnice D-Bus, systémů Samba a PolicyKit.

### 2.1 Správce souborů GNOME Nautilus

Podkapitola popisuje základní vlastnosti programu Nautilus včetně možných způsobů jeho rozšíření. Data byla čerpána ze zdroje [1].

Nautilus je intuitivní a silný souborový manažer prostředí GNOME. Díky využití GVFS [2] je Nautilus schopen zobrazit vzdálené složky stejně jako složky lokálního souborového systému. Dává tím uživatelům možnost přistupovat k jejich souborům za použití stejného rozhraní.

Nautilus umožňuje vývojářům využít zásuvných modulů k rozšíření funkcionality a poskytnout tak více informací o souborech uživatele. Uživatel tím získá bohaté zkušenosti a odbourá se potřeba využít více různých aplikací pro správu odlišných typů souborů.

Zásuvné moduly mohou rozšířit funkcionalitu souborového správce Nautilus mnoha rozdílnými způsoby. Umožňují například přidat další informace do dialogového okna vlastností, které poskytuje uživateli vše, co potřebuje vědět o vybraném souboru. Zásuvné moduly poskytují programátorům možnost přidat sloupce zobrazující informace o souboru či adresáři, viditelné, je-li Nautilus přepnut do Zobrazení v seznamu. Moduly lze využít pro automatické vkládání emblémů na soubory a tím poskytnout uživateli patřičné informace na první pohled.

Nautilus také dovoluje vývojářům upravovat kontextové menu souboru a složky. Položky kontextového menu mohou mít určeny typy souborů, u kterých budou zobrazeny. Tímto způsobem kontextové menu poskytuje pouze akce, které jsou určeny pro vybraný typ souboru.

Kromě flexibilní architektury zásuvných modulů, Nautilus umí zobrazit i náhled jakéhokoli typu souboru. Náhled usnadňuje uživateli najít soubory, které potřebuje. Prostředí GNOME umí pomocí tzv. thumbnail aplikace automaticky vytvořit náhled pro většinu obrazových a video souborů a zároveň dovoluje aplikacím nainstalovat program pro vytváření náhledů souborů specifických pro danou aplikaci.

### 2.1.1 Doplněk souborového správce Nautilus

Nautilus může být rozšířen za pomoci skriptů a za pomoci doplňků. Skripty jsou typicky jednodušší než doplňky a mohou být napsány v jakémkoli skriptovacím jazyce, který je daný operační systém schopný provést [3].

Doplňky souborového správce Nautilus poskytují více svobody jak Nautilus obohatit o další funkce. Pomocí doplňků lze implementovat přidávání emblémů, položek kontextového menu, sloupců při Zobrazení v seznamu a záložky v okně vlastností. Doplněk má podobu dynamické knihovny využívající Nautilus Extension API [4] a je typicky instalován administrátorem systému [5].

## 2.2 Sdílení souborů systémem Samba

Tato podkapitola byla převzata z [6] a jsou v ní uvedeny obecné informace o systému Samba společně se stručným popisem vlastností a chování tohoto systému.

Samba je svobodná implementace síťového protokolu SMB (Server Message Block) vydávána pod licencí GNU GPL. Lze spustit na většině Unix systémech i na systémech založených na Unixu, jako je Linux, Solaris, AIX a BSD varianty včetně Mac OS X od firmy Apple. Stala se standardem skoro ve všech Linuxových distribucích a je často zahrnuta do základní části systému.

Samba dovoluje sdílet soubory a tiskárny mezi počítači s operačním systémem MS Windows a počítači, na kterých je spuštěn Unix. Implementuje spousty služeb a protokolů včetně NetBIOS over TCP/IP, SMB, CIFS, DCE/RPC, WINS (NetBIOS Name Server), atd.

Samba nastavuje sdílení pro vybraný adresář včetně jeho podadresářů. Takto nasdílený adresář se zobrazí uživateli Microsoft Windows jako běžná složka přístupná přes síť. Uživatelé systému Unix mohou připojit sdílený adresář přímo do zvoleného adresáře za pomoci příkazu `smbmount`.

Služby, které Samba poskytuje jsou implementovány jako dva démoni:

- `smbd` pro sdílení souborů a tiskáren,
- `nbmd` poskytující službu NetBIOS-to-IP-adress name.

Nastavení Samby lze dosáhnout editací jednoho konfiguračního souboru `smb.conf` nebo použitím webového administračního rozhraní SWAT (Samba Web Administration Tool).



## 2.2.1 Konfigurační soubor smb.conf

K vytvoření této podkapitoly, která popisuje strukturu konfiguračního souboru systému Samba a pravidla pro definování sekcí sdílení, bylo využito zdrojů [7] a [8].

Konfigurační soubor Samby má výchozí název `smb.conf`. Má stejný formát jako Windows `.ini` soubory. Tento soubor obsahuje sekce a parametry. Sekce vždy začíná názvem v hranatých závorkách a pokračuje dokud nezačne jiná sekce nebo není konec souboru.

Sekce obsahuje parametry ve formátu:

```
jméno = hodnota
```

Konfigurační soubor je řádkově orientovaný, to znamená, že každý řádek zakončený znakem nového řádku představuje komentář, název sekce nebo parametr. Přičemž parametry a názvy sekcí nerozlišují malá a velká písmena (angl. case insensitive). Pouze první znak „=” (rovnítko) v parametru je důležitý. Bílé znaky před i za rovnítkem jsou zahozeny. Nepodstatné jsou také bílé znaky v názvu sekce a jméně parametru. Začáteční a koncové bílé znaky hodnoty parametru jsou také odstraněny. Jakýkoli řádek začínající středníkem („;“) nebo mřížkou („#“) je ignorován, stejně jako řádek obsahující pouze bílé znaky.

Každá sekce konfiguračního souboru (vyjma sekce `[global]`) popisuje sdílený zdroj. Parametry v této sekci definují vlastnosti sdílení. Existují tři speciální sekce, `[global]`, `[homes]` a `[printers]`. Parametry sekce `[global]` se aplikují na server jako celek nebo nastavují výchozí hodnoty parametrů, které nejsou definovány v dalších sekcích. Pokud konfigurační soubor obsahuje sekci `[homes]` server se snaží připojit klientům jejich domovský adresář. Nastavení sdílení tiskáren je možné v sekci `[printers]`.

Ukázka nastavení speciálních sekcí v konfiguračním souboru `smb.conf`:

```
[global]
  workgroup = TESTGROUP
  server string = Test-server
  security = share
  cups options = raw
  max log size = 50
```

```
[homes]
```

```
comment = Home Directories  
browseable = no  
writable = yes
```

```
[printers]
```

```
comment = All Printers  
path = /var/spool/samba  
browseable = no  
printable = yes
```

### **Sekce sdílení**

Sekce, která nezačíná názvem speciální sekce, popisuje sdílený zdroj a parametry definující vlastnosti sdílení. Mezi základní parametry pro nastavení sdílení patří parametry `path`, `comment`, `read only`, `writable` a `guest ok`.

Parametr `path` udává cestu ke sdílenému adresáři. Můžete si vybrat jakýkoli adresář na Samba serveru, pokud má vlastník Samba procesu, který se do adresáře připojuje, povolen zápis a čtení adresáře.

Parametr `comment` umožňuje napsat komentář, který bude poslán klientovi při prohlížení sdílení. Uživatel tento komentář může vidět v detailním zobrazení sdílené složky nebo použitím příkazu `net view` v příkazové řádce MS-DOS.

Parametry `read only` a `writable` (také `writeable` nebo `write ok`) jsou dva různé způsoby určující stejnou věc. Definují oprávnění čtení a zápisu do sdílené složky. Tyto parametry jsou výlučné a proto lze při nastavování oprávnění přístupu do sdílené složky využít libovolný z nich.

Parametr `guest ok` říká, zda bude požadováno heslo při připojení do složky uživatelem ze vzdáleného počítače.

Následující tabulka (tabulka 1) shrnuje jména všech základních parametrů sekce konfiguračního souboru systému Samba společně s výchozími hodnotami.

Název parametru	Typ hodnoty parametru	Výchozí nastavení
path	Řetězec (cesta k adresáři)	<i>/tmp</i>
comment	Řetězec	<i>žádné</i>
read only	boolean	<i>yes</i>
writable	boolean	<i>no</i>
guest ok	boolean	<i>no</i>

Tabulka 1: Základní parametry sdílené sekce

Ukázka nastavení sdílení adresáře Projects v konfiguračním souboru `smb.conf`:

```
[Work]
    path = /home/User/Projects
    comment = work projects
    read only = no
    writable = yes
    guest ok = no
```

## 2.3 Sběrnice D-Bus

D-Bus (angl. Desktop bus) je jednoduchý systém sloužící pro meziprocesovou komunikaci (angl. IPC – inter-process communication). Tento svobodný software tak poskytuje aplikacím jednoduchý způsob vzájemné komunikace. Komunikace probíhá přes D-Bus démona. Současně může být spuštěno i několik instancí neboli kanálů. Nejčastější instancí je systémová instance, která je jedinečná v rámci celého systému a jsou na ni kladena přísná bezpečnostní omezení. Z tohoto důvodu jsou potřebné privátní instance, které jsou vytvořeny pro každého přihlášeného uživatele, umožňující komunikaci bez omezení [9].

Prostředí GNOME resp. aplikace, které jsou zahrnuty do tohoto desktopového prostředí, vzájemně komunikují přes sběrnici D-Bus. V současné době D-Bus využívá celá řada systémů založených na Unixu. Mezi ně patří i operační systém Fedora vyvíjený komunitou a firmou Red Hat.

Systém D-Bus se skládá ze tří vrstev. První vrstvou je knihovna `libdbus`, která umožňuje spojení dvou aplikací a vzájemnou výměnu zpráv. Spustitelný démon, postavený na knihovně `libdbus`, ke kterému se mohou aplikace připojovat, je vrstvou druhou. Démon umožňuje doručit zprávy

poslané z jedné aplikace žádné nebo více aplikacím. Poslední vrstvou jsou obalující knihovny (angl. wrapper libraries) nebo vazby (angl. bindings) založené na aplikačních rámcích. D-Bus byl navržen pro komunikaci mezi desktopovými aplikacemi v rámci stejného desktopového sezení a pro komunikaci mezi desktopovým sezením a operačním systémem, kde operační systém typicky zahrnuje jádro, systémové démony a systémové procesy [10].

Každá aplikace, která používá systém D-Bus obsahuje objekty. Tyto objekty přijímají zprávy zaslané přes sběrnici D-Bus. Každý objekt má své jedinečné jméno mající tvar cesty v souborovém systému například: `/org/fedoraproject/SimpleFileShare` [9].

## 2.4 Systém PolicyKit

PolicyKit je nástroj pro definování a obsluhu oprávnění, které povoluje neoprávněným procesům komunikovat s procesy oprávněnými. Tento aplikační rámec centralizuje rozhodovací proces s ohledem na povolení přístupu k oprávněné operaci vyžádané neoprávněnou aplikací. PolicyKit se speciálně zaměřuje na aplikace v desktopovém prostředí na víceuživatelských operačních systémech založených na Unixu [11].

Dále pak ze zdroje [12] zjistíme, že jádro PolicyKitu je implementováno jako sdílená knihovna `libpolkit`. Představme si `libpolkit` jako černou skříňku, jejíž role je odpovědět, zdali má dotazující se program oprávnění vykonat danou akci.

Odpověď knihovny `libpolkit` není limitována pouze na typ boolean. Mohou být vráceny hodnoty:

- ano (angl. yes), program může pokračovat ve vykonání akce,
- ne (angl. no), program nemůže vykonávat akci,
- vyžaduje autentizaci (angl. require authentication), program musí zažádat uživatele o autentizaci, aby mohl pokračovat v provedení akce.

Autentizace může být specifikovaná jako autentizace uživatele (uživatel zadává svoje vlastní heslo) nebo jako autentizace super uživatele (uživatel musí zadat heslo super uživatele – root). Přičemž autentizace může být zachována natrvalo (přetrvá přes restart systému a je společná pro různé desktopové sezení) nebo je zachována pouze v rámci sezení, kterého je program součástí nebo pouze po dobu životního cyklu programu.

# 3 Specifikace požadavků

Kapitola stručně popisuje požadavky, které musí být dodrženy při návrhu a implementaci projektu. Rozebírá jednotlivé části z pohledu uživatele i z pohledu programátora.

Obecné požadavky:

- vytvoření doplňku souborového správce Nautilus,
- vytvoření uživatelského rozhraní umožňující nastavení parametrů sdíleného adresáře a komunikaci s démonem,
- vytvoření aplikace typu démon poskytující funkce pro:
  - zjištění informací o sdíleném adresáři,
  - vytvoření sdílení zvoleného adresáře,
  - změnu parametrů sdíleného adresáře,
  - zrušení sdílení adresáře,
  - načítání potřebných informací z konfiguračního souboru Samby,
- spuštění démona s oprávněním modifikace konfiguračního souboru Samby,
- zajištění komunikace klientské části a démona přes sběrnici D-Bus,
- kontrola oprávnění zápisu do konfiguračního souboru Samby za pomoci PolicyKit subsystému.

Jedním z požadavků je vytvoření doplňku souborového správce Nautilus, který spustí grafické uživatelské rozhraní. Doplněk musí:

- být vytvořen jako dynamická knihovna,
- používat Nautilus Extension API,
- rozšířit správce souborů Nautilus.

Jak již bylo zmíněno, dalším bodem je vytvoření uživatelského rozhraní umožňující nastavení parametrů sdíleného adresáře a komunikaci s démonem. Na něj jsou kladeny požadavky:

- použití v prostředí GNOME,
- lehce rozšiřitelné,
- možnost lokalizace,
- použití ikon a témat prostředí,
- kontrola oprávnění uživatele,
- komunikace přes systémovou sběrnici D-Bus.

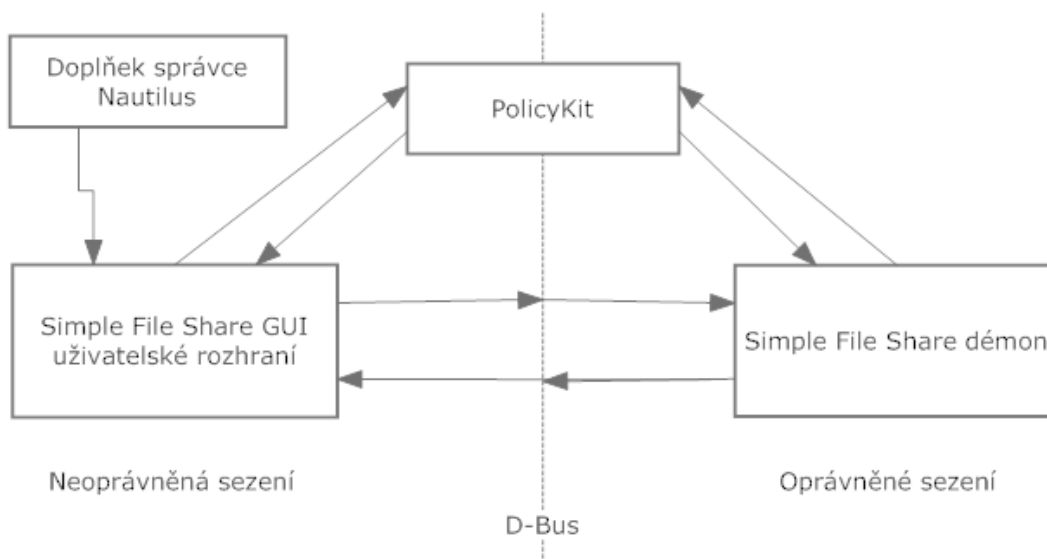
Aplikace typu démon je poslední celistvou částí projektu. Musí poskytovat funkce pro práci s konfiguračním souborem Samba systému. Program musí být:

- konzolová aplikace,
- lehce rozšiřitelný,
- snadno ovladatelný,
- schopen kontrolovat oprávnění příchozích požadavků,
- registrován na systémové sběrnici D-Bus.

Z výše zmíněných požadavků vyplývají různá omezení na softwarové vybavení počítače. Operační systém založený na Unixu s desktopovým prostředím GNOME a knihovny GTK+ je nezbytný. Musí být nainstalován systém D-Bus potřebný pro komunikaci mezi aplikacemi a systém PolicyKit, pomocí kterého bude zajištěna kontrola oprávnění k zápisu do konfiguračního souboru systému Samba.

## 4 Návrh řešení projektu

Projekt se skládá ze tří částí. První částí je doplněk souborového správce Nautilus. Tento doplněk slouží ke spuštění grafického uživatelského rozhraní, které komunikuje s démonem. Z obrázku 4.1 lze vyčíst, že démon je samostatný program, který odpovídá a provádí požadavky zaslány klientem přes sběrnici D-Bus.



Obrázek 4.1: Návrh komunikace jednotlivých částí

### 4.1 Doplněk souborového správce Nautilus

Doplněk bude vytvořen jako dynamická knihovna využívající Nautilus Extension API. Úkolem doplňku je přidat položku do kontextového menu resp. do nabídky Úpravy (angl. Edit).

Mohou nastat tři různé situace, při kterých se zobrazuje kontextové menu:

- uživatel označí soubor a zobrazí si menu; jelikož nelze sdílet pouze soubor, nebude do žádné z nabídek přidáno tlačítko,
- uživatel označí skupinu více položek (adresářů, souborů); i v tomto případě se tlačítko sdílení do žádné z nabídek nepřidá. Důvodem je nejednoznačnost. Nemůžeme určit, kterou z vybraných položek si uživatel přeje sdílet,
- uživatel označí vybraný adresář; nejedná se o soubor a výběr je jednoznačný. V tomto případě doplněk souborového správce Nautilus přidá tlačítko do kontextového menu vybraného adresáře i do nabídky Úpravy.

Kliknutím na přidání tlačítka Nautilus zavolá obsluhu tlačítka popsanou v navrženém doplňku. V této obsluze se zjistí cesta k uživatelem zvolenému adresáři. Tato cesta se předá jako první parametr při spouštění grafického uživatelského rozhraní.

## 4.2 Grafické uživatelské rozhraní

Nautilus se v GNOME používá jako výchozí aplikace pro manipulaci se soubory. GNOME je prostředí pracovní plochy v unixových operačních systémech. Toto pracovní prostředí je postaveno nad knihovnou GTK+ (The GIMP Toolkit), která byla napsána pro bitmapový editor GIMP. Vytvořené grafické uživatelské rozhraní bude, ze zřejmých důvodů, také využívat knihovnu GTK+.

Uživatelské rozhraní má být jednoduché, intuitivní a přehledné. Je třeba, aby se uživatel rychle zorientoval v jeho ovládání. Od aplikace je požadováno předpokládat a ošetřovat různé chybové situace.

### 4.2.1 Dialogové okno aplikace

Okno aplikace musí dát uživateli na výběr, zda chce vybranou složku sdílet pomocí systému Samba (Windows share) a nebo si přeje vybranou složku nesdílet resp. zrušit nastavené sdílení. Přeje-li si uživatel zvolenou složku sdílet, musí mít k dispozici interaktivní objekty, pomocí kterých bude schopen nastavit všechny základní parametry sdíleného adresáře. Pro určení základních vlastností sdíleného adresáře dialogové okno musí obsahovat:

- editační pole pro zadání názvu sdílené složky,
- editační pole pro možnost zadání nepovinného komentáře sdílené složky,
- zaškrtačkové políčko určující, zda složka bude pouze pro čtení nebo bude povolen i zápis do zvolené složky,
- zaškrtačkové políčko povolující přístup hosta do sdílené složky.

K uložení provedených změn nesmí dialogové okno postrádat tlačítko Save, které odešle změny ke zpracování. Nastane-li situace, že se uživatel rozhodne neukládat nastavené změny, může kdykoli aplikaci ukončit kliknutím na tlačítko Close.

### 4.2.2 Komunikace přes sběrnici D-Bus

Klientská část bude získávat informace o nastavení sdílení adresáře přes systémovou sběrnici D-Bus. Stejnou cestou budou odesílány i údaje zadané uživatelem. Pokud tuto komunikaci rozebereme podrobněji zjistíme, že aplikace bude volat tři různé metody registrované na sběrnici D-Bus. První



z nich je metoda, pomocí které démon při odpovědi zašle všechny základní parametry popsané v sekci sdílení adresáře. Pokud vybraný adresář není dosud sdílený bude odpověď prázdná. Další metoda slouží k odeslání vyplněných údajů resp. vlastností sdílení adresáře. Tato metoda je volána asynchronně, protože odpověď nemusí být doručena ihned. Poslední metodou je metoda, pomocí které můžeme zrušit sdílení adresáře. Stejně jako předchozí je tato metoda volána asynchronně. Odpovědi na tyto metody mohou být při úspěšném provedení kladné, při neúspěchu je přijata odpověď s popisem vzniklé chyby.

### 4.2.3 PolicyKit – kontrola oprávnění

Konfigurační soubor Samby má možnost měnit pouze administrátor systému. Z tohoto důvodu se musí kontrolovat oprávnění uživatele, který spustil aplikaci. Kontrola se bude provádět pouze před použitím jedné z metod registrovaných na sběrnici D-Bus, jejímž výsledkem je změna obsahu konfiguračního souboru systému Samba. Tyto metody jsou volány při změně parametrů sdílení a při rušení sdílení adresáře. Z toho vyplývá, že před voláním jedné ze zmíněných metod se musí aplikace dotázat autentizačního agenta zda má dostatečná oprávnění. Pokud oprávnění má, pokračuje voláním příslušné metody. Pokud autentizační agent zjistí, že aplikace nemá patřičná práva, zobrazí dialog vyzývající uživatele o zadání hesla. Při úspěšné autentizaci se zapíše informace do databáze o úspěšném přidělení oprávnění aplikace.

## 4.3 Simple File Share démon

Démon je obecně program, který je spuštěn na pozadí a není v přímém kontaktu s uživateli. Čeká na konkrétní událost. Jakmile taková událost nastane, démon ji obslouží a po její obsluze dále vyčekává v nečinnosti.

### 4.3.1 Práce s konfiguračním souborem

Pro provádění změn v konfiguračním souboru systému Samba je nutné, aby byl démon spuštěn s právy dovolující editaci souboru `smb.conf`. Takováto práva získá, spustíme-li jej pod účtem administrátora systému.

Démon musí poskytovat tři základní funkce, pracující s konfiguračním souborem:

- načtení údajů o zvoleném adresáři,
- zapsání nového sdílení či změnu údajů v sekci vybraného adresáře,
- zrušení celé sekce, odstranění sdílení.

### **Načtení sekcí z konfiguračního souboru**

K této operaci není třeba mít oprávnění. Démon otevře `smb.conf` pouze pro čtení. Projde celý soubor a vyhledává všechny sekce popisující sdílení. Speciální sekce a jejich parametry jsou ignorovány. U všech ostatních sekcí si démon zapamatuje jejich název a základní parametry jimiž jsou: cesta ke sdílenému zdroji, komentář, údaj, zda je složka pouze pro čtení resp. údaj, zda je do složky povoleno zapisovat a poslední údaj povolující účet hosta.

### **Načtení údajů o zvoleném adresáři**

Při požadavku o získání parametrů démon jednoduše projde načtené sekce z konfiguračního souboru. Vyhledá, zda je cesta ke zvolenému adresáři shodná s parametrem `path` jedné ze sekcí. Pokud najde shodu, odpoví na požadavek zasláním načtených parametrů.

### **Vytvoření nové sekce**

Démon musí umět obsloužit požadavek žádající o vytvoření nové sekce. Nejdříve se ujistí, jestli neexistuje sekce se stejným názvem. Poté připojí na konec konfiguračního souboru novou sekci s hodnotami parametrů, které uživatel zvolil v grafickém uživatelském rozhraní.

### **Změna parametrů existující sekce**

Démon při požadavku změny nastavení sdíleného adresáře načte konfigurační soubor. Zjistí zda je zvolený adresář opravdu sdílený. Je-li sekce zapsaná v `smb.conf`, démon přepíše jednotlivé hodnoty základních parametrů v souboru hodnotami, které uživatel změnil v grafickém uživatelském rozhraní.

### **Odstranění sekce**

Obsluha požadavku o odstranění sekce začíná načtením sekcí z konfiguračního souboru. Poté démon testuje, zda se shoduje parametr `path` jedné ze sekcí s uživatelem vybranou cestou. Při shodě tuto sekci odstraní z konfiguračního souboru `Samby` a tím zruší sdílení zvoleného adresáře.

## **4.3.2 Komunikace přes sběrnici D-Bus**

Démon poskytuje tři metody registrované na systémové sběrnici D-Bus. Tyto metody jsou:

- `get_share_status`; vracející nastavené vlastnosti sdíleného adresáře,
- `setup_share`; metoda zavolá vytvoření sekce resp. změnu parametrů existující sekce,
- `delete_share`; metoda zavolá funkci odstraňující sekci z konfiguračního souboru.

Metoda `get_share_status` je metoda synchronní. Při obsluze této metody se provede načtení údajů o zvoleném adresáři. V případě bezchybného načtení odpovídajících údajů se tyto údaje odešlou v parametru obsluhované metody. Nebyl-li adresář nalezen v konfiguračním souboru, odešle se ve výstupním parametru prázdné pole. O vzniku chyby je uživatelské rozhraní informováno nastavenou hodnotou v chybovém parametru.

Oproti `get_share_status` je `setup_share` metodou asynchronní. Metoda `setup_share` se stará o zavolání funkce pro vytvoření nové sekce (nového sdílení) a nastavení základních parametrů a také o provedení funkce provádějící změnu základních parametrů sdílené složky. Při neúspěchu je uživatel informován nastavenou chybou v odpovědi na požadavek.

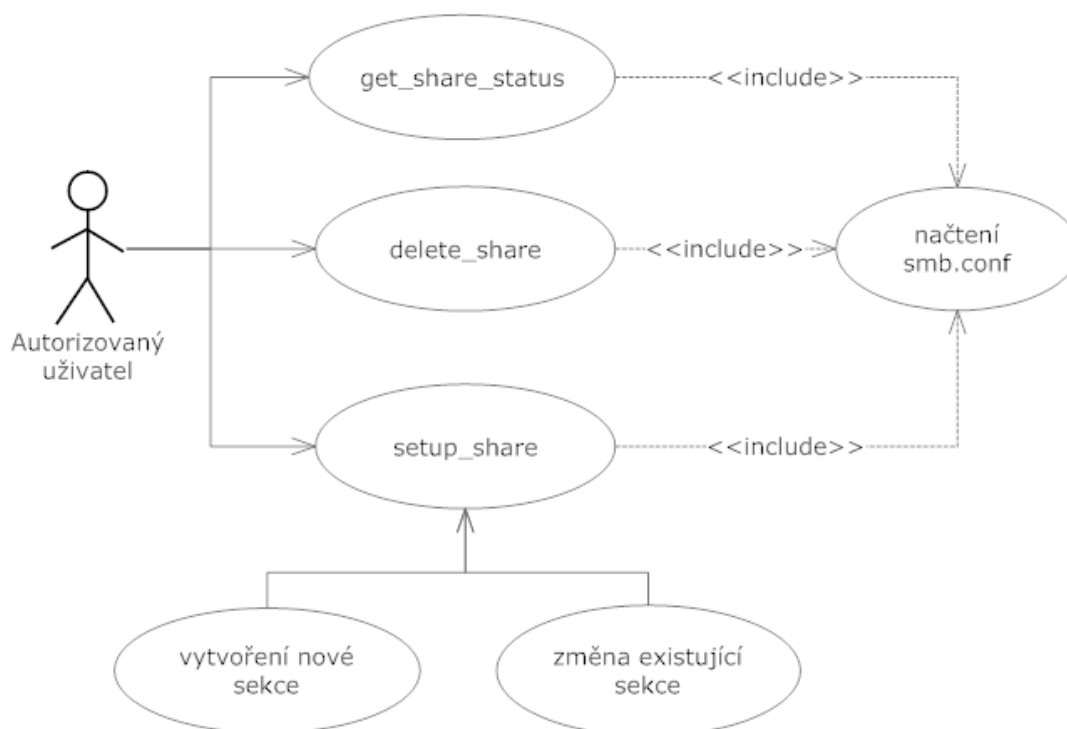
Stejně jako předchozí metoda je `delete_share` metoda asynchronní. Jejím úkolem je spustit funkci, která odstraní celou sekci z konfiguračního souboru. Podobně jako v předchozích případech je uživatelské rozhraní informováno o nastalé chybě nastavením parametru chyby.

### **4.3.3 PolicyKit – kontrola oprávnění**

Kontrola, zda aplikace, která zavolala jednu z metod `setup_share` a `delete_share`, má oprávnění k provádění změn v konfiguračním souboru, je provedena za pomoci PolicyKit systému. Démon nahlédne do autentizační databáze, která uchovává údaje o úspěšné autentizaci. Pokud zjistí, že aplikace žádající obsluhu má dostatečná oprávnění, pokračuje v provádění volané metody. Není-li v autentizační databázi záznam, nemá aplikace dostačující oprávnění. V tomto případě démon nepokračuje v obsluze zvolené metody, nýbrž odešle odpověď s nastavenou chybou značící nedostatečné oprávnění k použití volané metody.

## 4.4 Diagram a specifikace případů užití

Z obrázku 4.2 je patrné, že uživatel bude využívat pouze tři funkce, které budou poskytovány démonem. Všechny funkce obsahují volání funkce načítající sdílené sekce z konfiguračního souboru `smb.conf`. Jedna z metod je zobecněním dvou funkcí. Konkrétně je to funkce pro vytvoření nové sekce a funkce pro změnu základních parametrů existující sekce.



Obrázek 4.2: Diagram případu užití pohledem autorizovaného uživatele.

Všechny metody znázorněné v diagramu případu užití (obrázek 4.2) jsou blíže popsány a jejich chování je více rozebráno v následujících specifikacích případu užití. Tyto specifikace slovně popisují chování jednotlivých metod a navrhnou možné reakce na vzniklé události.

<b>Případ užití: get_share_status</b>
<p><b>Účastníci:</b> Uživatel</p>
<p><b>Tok událostí:</b></p> <ol style="list-style-type: none"> <li>1. Uživatel ve správci Nautilus zvolí v menu vybraného adresáře položku „<i>Share folder</i>“</li> <li>2. Před zobrazením dialogového okna nastavení sdílení se zavolá <code>get_share_status</code>, kde absolutní cesta k vybranému adresáři bude vstupním parametrem</li> <li>3. Provede se načtení konfiguračního souboru <code>smb.conf</code></li> <li>4. KDYŽ nebylo nalezeno sdílení adresáře: <ol style="list-style-type: none"> <li>4.1. Ve výstupním parametru vrátím prázdné pole</li> </ol> </li> <li>5. Ve výstupním parametru vrátím pole hodnot základních parametrů sdílení, které byly přečteny z konfiguračního souboru <code>smb.conf</code></li> </ol>
<p><b>Následné podmínky:</b></p> <ol style="list-style-type: none"> <li>1. Informace předané výstupním parametrem se zobrazí v dialogu nastavení sdílení</li> </ol>

<b>Případ užití: setup_share</b>
<p><b>Účastníci:</b> Uživatel</p>
<p><b>Vstupní podmínky:</b></p> <ol style="list-style-type: none"> <li>1. Uživatel má oprávnění ke změně <code>smb.conf</code></li> </ol>
<p><b>Tok událostí:</b></p> <ol style="list-style-type: none"> <li>1. Uživatel vyplní povinné údaje v dialogovém okně pro nastavení sdílení</li> <li>2. Stisknutím tlačítka <code>Save</code>, se zavolá metoda <code>setpu_share</code>, která předá uživatelem nastavené hodnoty základních parametrů sdílení ve vstupním parametru volané metody.</li> <li>3. Provede se načtení konfiguračního souboru <code>smb.conf</code></li> <li>4. KDYŽ nebylo nalezeno sdílení adresáře: <ol style="list-style-type: none"> <li>4.1. Nastaví sdílení adresáře s příslušnými parametry v <code>smb.conf</code></li> </ol> </li> <li>5. Změní parametry sdíleného adresáře za nově zadané uživatelem</li> </ol>
<p><b>Následné podmínky:</b></p> <ol style="list-style-type: none"> <li>1. Bylo vytvořeno resp. změněno sdílení zvoleného adresáře</li> <li>2. Znovunačtení konfiguračního souboru <code>smb.conf</code> Samba démonem</li> </ol>
<p><b>Alternativní tok:</b></p> <ol style="list-style-type: none"> <li>1. Uživatel může kdykoli zavřít dialogové okno</li> </ol>

<b>Případ užití: delete_share</b>
<p><b>Účastníci:</b></p> <p>Uživatel</p>
<p><b>Vstupní podmínky:</b></p> <ol style="list-style-type: none"> <li>1. Uživatel má oprávnění ke změně smb.conf</li> </ol>
<p><b>Tok událostí:</b></p> <ol style="list-style-type: none"> <li>1. Uživatel v dialogovém okně nastaví parametr sdílení na „<i>Do not share this folder</i>“</li> <li>2. Stisknutím tlačítka Save, se zavolá metoda delete_share, kde absolutní cesta k vybranému adresáři bude vstupním parametrem.</li> <li>3. Démon provede načtení konfiguračního souboru smb.conf</li> <li>4. KDYŽ bylo nalezeno sdílení adresáře: <ol style="list-style-type: none"> <li>4.1. Zruší sdílení adresáře odstraněním záznamu v smb.conf</li> </ol> </li> </ol>
<p><b>Následné podmínky:</b></p> <ol style="list-style-type: none"> <li>1. Bylo zrušeno sdílení zvoleného adresáře</li> <li>2. Znovunačtení konfiguračního souboru smb.conf Samba démonem</li> </ol>
<p><b>Alternativní tok:</b></p> <ol style="list-style-type: none"> <li>1. Uživatel může kdykoli zavřít dialogové okno</li> </ol>

## 5 Implementace projektu

K implementaci zásuvného modulu, grafického uživatelského rozhraní i démona bylo použito svobodné multiplatformní vývojové prostředí Code::Blocks. Obě aplikace i zásuvný modul jsou napsány v jazyce C standardu C89. Zdrojové kódy byly překládány překladačem GCC (GNU Compiler Collection) verze 4.4.3. Operačním systémem použitým při vývoji byla Fedora 12 (2.6.32.11-99.fc12.x86\_64).

### 5.1 Zásuvný modul správce Nautilus

Zásuvný modul souborového správce Nautilus je popsán v souboru `nautilus_sfshare.c`. V tomto souboru, konkrétně ve funkci `sfshare_extension_register_type`, je implementováno zaregistrování nového typu a přidání rozhraní. V další funkci s názvem `sfshare_extension_get_file_items` je prováděna kontrola dvou podmínek. První z nich je kontrola výběru pouze jedné položky. Pokud je podmínka splněna, pokračuje se vyhodnocením druhé podmínky. Ta říká, že pokud vybraná položka není adresář dále, nesmím pokračovat. Jsou-li obě podmínky splněny, je v těle `nautilus_menu_item_new` přidána nová položka kontextového menu reagující na její aktivaci provedením kódu funkce `sfshare_start_gui`. Tento kód pomocí funkce `g_spawn_async` spustí grafické uživatelské rozhraní s jedním parametrem. Tím parametrem je cesta ke složce, kterou označil uživatel.

### 5.2 Grafické uživatelské rozhraní

Aplikace bude používána především v prostředí pracovní plochy GNOME, z toho vyplývá omezení na použití knihovny GTK+. Lze ji rozdělit na dvě části obstarávající odlišné funkce. V první části je implementováno grafické rozhraní, tj. vytvoření dialogového okna, obsluha tlačítek a zpracování signálů. V druhé části jsou implementovány funkce pro odesílání a příjem zpráv přes systémovou sběrnici D-Bus a také ověřování oprávnění využívající systém PolicyKit.

Soubor `sfshare.c` je hlavním souborem grafického uživatelského rozhraní obsahující funkci `main`. V této funkci se provádí kontrola správně zadaného parametru definujícího cestu ke zvolenému adresáři. Pokud není počet parametrů při spuštění roven jedné, je ohlášena chyba a aplikace ukončena. Následně je vytvořeno hlavní okno (obrázek 5.1) spolu s jeho obsahem: výsuvnou nabídkou, tlačítka `Save` a `Close`, editovacími poli i zaškrtnávacími tlačítka.

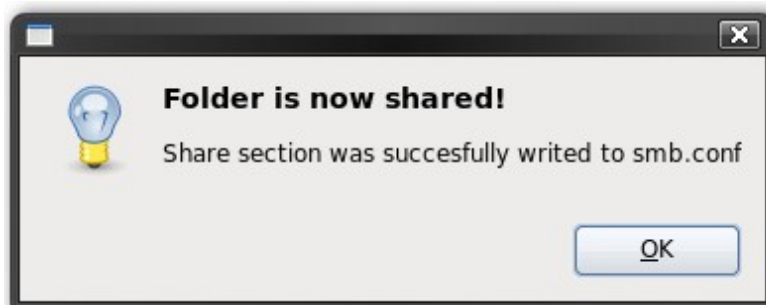


Obrázek 5.1: Dialogové okno programu Simple File Share

Rozmístění komponent dialogového okna, je popsáno v xml souboru `sfshare_window.ui`. Funkcí `gtk_builder_new` naplním proměnnou typu `GtkBuilder`. Do této proměnné je přidán obsah souboru `sfshare_window.ui`. Následně jsou volány funkce `gtk_builder_get_object`, které přiřadí hodnoty jednotlivým proměnným typu `GtkWidget` deklarovaných ve struktuře `SFSDialog`. Ke všem komponentám se může připojit reakce na určitou událost. K tomuto účelu slouží funkce `g_signal_connect`. Hlavní okno grafického uživatelského rozhraní je zobrazeno funkcí `gtk_widget_show` a následně spuštěno při zavolání funkce `gtk_main`.

V souboru `sfshare.c` je mimo jiné implementována funkce `show_message`. Tato funkce slouží ke snadnému vytvoření a zobrazení modálního varovného (obrázek 5.3), chybového nebo informativního okna (obrázek 5.2) spolu s doplňující zprávou. Tělo funkce obsahuje vytvoření dialogu pomocí funkce `gtk_message_dialog_new`. Doplňující text je vytvořen funkcí `gtk_message_dialog_format_secondary_text`. Celé okno zprávy je zobrazeno jako modální zavoláním funkce `gtk_dialog_run`.





Obrázek 5.2: Informativní zpráva



Obrázek 5.3: Varovná zpráva

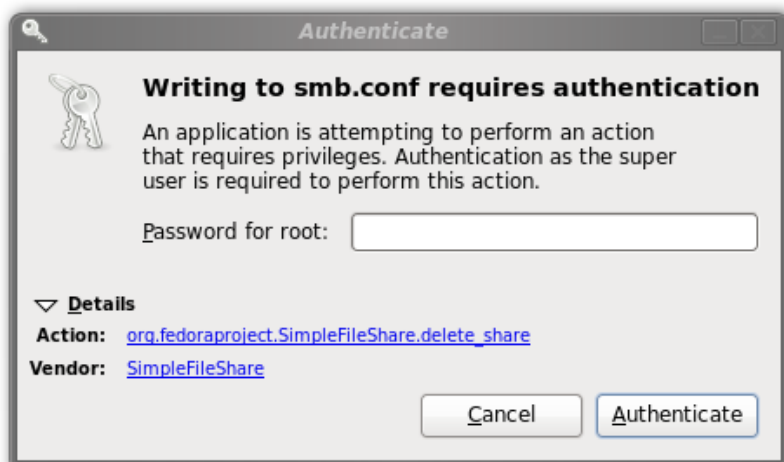
Pokud je tlačítko `Save` aktivní, může jej uživatel stisknout a zavolat tak funkci `sfshare_authority_check` ověřující oprávnění. Má-li aplikace dostatečná oprávnění, pokračuje prováděním těla funkce `save_share`. Funkce zjistí typ akce, kterou má provést. Žádná akce se neprovede, není-li adresář doposud sdílený a uživatel má vybranou volbu „*Do not share this folder*“. Je-li vybraná volba „*Do not share this folder*“ a existuje-li sekce v konfiguračním souboru, ve které je popsáno sdílení adresáře, je zavolána funkce `dbus_sfshare_delete_share`, která se pokusí sekci odstranit. Pokud uživatel vybral volbu „*Windows Share [Samba]*“ zjistí se všechny nastavené parametry a předají se funkci `dbus_sfshare_set_share`.

Další část je popsána v souboru `dbus_client.c`. V tomto souboru jsou funkce komunikující s démonem za pomoci D-Bus a současně jsou zde i funkce kontrolující oprávnění systémem PolicyKit.

Zavoláním funkce `dbus_sfshare_connect` se otestuje, jestli je klientská aplikace připojena na sběrnici D-Bus. Není-li připojen, pokusí se aplikace spojit s D-Bus. Nastane-li chyba, je o ní uživatel informován pomocí modálního okna vytvořeného funkcí `show_message`. Opakem k této funkci je funkce `dbus_sfshare_disconnect`, která připojenou aplikaci odpojí od sběrnice D-Bus.

Podobně jako funkce pracující s D-Bus, jsou zde i funkce PolicyKit. Inicializaci PolicyKit autority zajišťuje funkce `polkit_sfshare_init`. Při ukončení aplikace se volá funkce `polkit_sfshare_free`, která odstraní objekty PolicyKit.

Chceme-li zjistit oprávnění aplikace, které je potřebné k používání metod registrovaných na sběrnici D-Bus, použijeme asynchronní funkci `polkit_sfshare_check`. Při volání funkce je nutné zadat parametr, kterým se určí `action_id`. Typ `action_id` určuje metodu, pro kterou se má oprávnění nastavit. Z autorizační databáze zjistí, zda je aplikace oprávněna volat zadanou metodu. Pokud nemá dostatečná oprávnění, PolicyKit zobrazí okno, vyzývající uživatele k zadání hesla administrátora systému (obrázek 5.4). Vyhodnocení probíhá ve funkci `check_authorization_cb`. Zadal-li uživatel správné heslo, pokračuje se v provádění přerušené operace. V tomto případě je to reakce na událost stisku tlačítka `Save`. Pakliže nebylo zadáno správné heslo nebo uživatel přerušil výzvu k zadání hesla, aplikace dále nepokračuje v obsluze tlačítka.



Obrázek 5.4: Výzva k autentizaci uživatele

Při spuštění aplikace se volá funkce `dbus_sfshare_get_share`. Ta implementuje volání synchronní metody `org_fedoraproject_SimpleFileShare_get_share_status`, která zjistí zda je vybraný adresář sdílený. Existuje-li sekce v konfiguračním souboru Samby s parametrem `path`, který je roven zadané cestě, obdrží klient odpověď. V odpovědi je seznam všech základních parametrů včetně názvu samotné sekce. Seznam bude prázdný v případě, že neexistuje žádná sekce popisující sdílení zvoleného adresáře.

Grafické uživatelské rozhraní používá dvě asynchronní metody registrované na sběrnici D-Bus. Jedna z těchto metod je volána ve funkci `dbus_sfshare_set_share`. Tato funkce volá metodu

`org_fedoraproject_SimpleFileShare_setup_share_async`, které jako vstupní parametr předá pole řetězců. Toto pole specifikuje hodnoty jednotlivých základních parametrů sekce. Odpověď metody je zpracována ve funkci `set_share_reply`, která zobrazí informaci o úspěšném zapsání sekce do konfiguračního souboru `smb.conf`. V případě chyby se zobrazí uživateli varování s textem popisujícím vzniklou chybu.

Druhá asynchronní metoda je volána ve funkci `dbus_sfshare_delete_share`. Metodu registrovanou na systémové sběrnici D-Bus voláme pomocí funkce `org_fedoraproject_SimpleFileShare_delete_share_async`. Vstupní parametr popisuje cestu ke zvolenému adresáři. Při volání funkce se zadává i tzv. callback. Callback je funkce, která je provedena při obdržení odpovědi. V tomto případě je to funkce `delete_share_reply`, jejíž obsah je podobný funkci `set_share_reply`. Nastane-li chyba je o ní uživatel informován zobrazeným varováním s textem popisujícím vzniklou chybu. Při úspěšném odstranění sekce z konfiguračního souboru Samby, se uživateli zobrazí okno informující o úspěchu.

## 5.2.1 PolicyKit v grafickém uživatelském rozhraní

Inicializace PolicyKit znamená získání autority. `PolkitAuthority` získáme návratovou hodnotou funkce `polkit_authority_get`. K ověření autentizace je zapotřebí znát také `PolkitSubject`. Tento subjekt získáme zavoláním funkce `polkit_system_bus_name_new`, na jejíž vstup je přivedena návratová hodnota funkce `dbus_bus_get_unique_name`, která jako parametr požaduje jedinečné jméno zveřejněné na sběrnici D-Bus. To lze získat funkcí `dbus_bus_get_unique_name`. Jako parametr je zde vloženo volání funkce `dbus_g_connection_get_connection`, které požaduje určení `DBusGConnection`. Proměnná tohoto typu je naplněna hodnotou získanou z funkce `dbus_g_bus_get`. Subjekt, společně s autoritou, názvem metody a dalšími parametry, je předán funkci `polkit_authority_check_authorization`. Ta zjistí, jestli je aplikace zapsaná v autentizační databázi. Výsledná hodnota, která je vrácená funkcí `polkit_authority_check_authorization_finish`, je dále testována. Pokud je `polkit_authorization_result_get_is_authorized` rovno `TRUE`, pak je aplikace autentizovaná a má přidělená dostatečná oprávnění pro změnu konfiguračního souboru. Pokud je test `polkit_authorization_result_get_is_challenge` roven `TRUE`, pak aplikace není autentizovaná, ale uživatel je vyzván k vyplnění hesla. Jsou-li oba testy rovny `FALSE`, pak aplikace není autentizovaná a nemá přidělena oprávnění.

## 5.2.2 Komunikace přes sběrnici D-Bus

Při pokusu o připojení zajistíme zadáním parametru `DBUS_BUS_SYSTEM`, že budeme komunikovat na systémové sběrnici D-Bus. Následně je vytvořen objekt poskytující metody. Jednoznačnými identifikátory objektu jsou: jméno, cesta a rozhraní objektu.

- `SFSHARE_DBUS_NAME "org.fedoraproject.SimpleFileShare"`
- `SFSHARE_DBUS_PATH "/org/fedoraproject/SimpleFileShare"`
- `SFSHARE_DBUS_INTERFACE "org.fedoraproject.SimpleFileShare"`

Pokud je aplikace připojena k objektu, je možné využít funkce implementované v souboru `dbus_client_glue.h`. Použité funkce jsou:

- `org_fedoraproject_SimpleFileShare_get_share_status;`  
zavolá metodu `get_share_status`,
- `org_fedoraproject_SimpleFileShare_setup_share_async;`  
zavolá asynchronní metodu `setup_share`,
- `org_fedoraproject_SimpleFileShare_delete_share_async;`  
zavolá asynchronní metodu `setup_share`.

Soubor `dbus_client_glue.h` byl vytvořen z xml souboru, který definuje názvy a parametry metod. Název zdrojového souboru je `org.fedoraproject.SimpleFileShare.xml`. Výsledný soubor je v jazyce C a byl vygenerován příkazem:

```
dbus-binding-tool --prefix=client --mode=glib-client \  
org.fedoraproject.SimpleFileShare.xml > dbus_client_glue.h
```

## 5.3 Implementace Simple File Share démona

Démon je spuštěn s právy administrátora systému. Jeho Hlavním úkolem je čekat na příchod požadavku od grafického uživatelského rozhraní. Následně požadavek obslouží a znovu čeká v nečinnosti. Veškerá komunikace klientské aplikace s démonem probíhá přes systémovou sběrnici D-Bus. Program nemá žádné grafické rozhraní a implementuje pouze funkce týkající se komunikace, ověřování oprávnění a editace konfiguračního souboru `smb.conf`.

Hlavní funkce, která je spuštěna jako první, je funkce `dbus_sfshare_start`. Nachází se v souboru `dbus_service.c`. Při volání funkce `dbus_g_object_type_install_info` jsou nainstalovány informace o objektu. Následně se získá spojení se systémovou sběrnici D-Bus funkcí

`dbus_g_bus_get` s parametrem `DBUS_BUS_SYSTEM`. Nastane-li chyba při pokusu o připojení, je oznámena uživateli a aplikace je ukončena. Pakliže proběhne připojení ke sběrnici D-Bus bezchybně, je zažádáno o zaregistrování jména `org.fedoraproject.SimpleFileShare`. Po získání jména je funkcí `g_object_new` vytvořen nový objekt. Tomuto objektu se registruje cesta `/org/fedoraproject/SimpleFileShare`. Mimo D-Bus funkce soubor `dbus_service.c` obsahuje také inicializaci `PolicyKit`. Za pomoci funkce `polkit_authority_get` je získána `PolicyKit` autorita. Proběhnou-li všechna tato volání bezchybně, je spuštěna hlavní smyčka programu příkazem `g_main_loop_run`. Démon je spuštěn a nečinně čeká na příchozí požadavky.

### 5.3.1 Obsluha příchozích požadavků

Démon má na systémové sběrnici D-Bus registrován objekt, který nabízí tři metody. Ty využívá klientská aplikace. Jednou z metod je metoda `daemon_get_share_status`. Je volána, když uživatel chce zjistit, jestli je adresář sdílený. Adresář je určen cestou předanou ve vstupním parametru metody. Ve druhém parametru je zaslána odpověď. Tento parametr je vyplněn ve funkci `smb_get_share_status`. Je-li parametr prázdný, pak adresář není sdílený. V opačném případě jsou nastaveny hodnoty základních parametrů sekce v pořadí: název sdílené sekce, cesta, komentář, údaj, zda je složka jen pro čtení a zda je povolen účet hosta. Pokud se v průběhu objeví chyba, je nastaven parametr typu `GError` a vrácena hodnota `FALSE`. Hodnota `TRUE` je vrácena, proběhne-li vše bezchybně.

První z metod, které jsou volány jako asynchronní, je metoda `daemon_setup_share`. V této funkci se zjišťuje voláním `polkit_authority_check_authorization`, zda má volaná aplikace oprávnění k zápisu do konfiguračního souboru `Samby`. Vyhodnocení je provedeno ve funkci `check_authorization_cb`. Pokud volající aplikace nemá dostatečná oprávnění, je poslána chyba funkcí `send_error`. Pokud existuje záznam v autorizační databázi, jsou splněny podmínky a je zavolána obsluha požadavku. V tomto případě je to funkce `setup_share_authorized`. Tělo obsahuje volání funkce `smb_set_share`, která vytvoří novou sekci nebo upraví parametry sekce stávající. V případě chyby vrací její hodnotu a ta je společně s textem odeslána pomocí funkce `send_error`. V případě, že žádná chyba nevznikla, je požádáno o znovunačtení konfiguračního souboru příkazem `smb_reload_service`. O úspěšném ukončení obsluhy je grafické uživatelské rozhraní informováno zavoláním funkce `dbus_g_method_return`.

Druhá metoda, kterou volající volá asynchronně, je metoda `daemon_delete_share`. Podobně jako u metody `daemon_setup_share` jsou i pro pokračování obsluhy vyžadována dostatečná oprávnění. Toho dosáhneme stejným způsobem jako u předchozí metody. Má-li uživatel udělena požadovaná

oprávnění, může být zavolána funkce `delete_share_authorized`. Autorizovaná obsluha obsahuje volání funkce `smb_delete_share`, která odstraní celou sekci z konfiguračního souboru `smb.conf`. Případná vzniklá chyba je předána uživateli zavoláním funkce `send_error`. Po úspěšném odstranění sdílení je požádáno funkcí `smb_reload_service` o znovunačtení souboru `smb.conf` Samba démonem. Ukončení obsluhy oznámíme zavoláním `dbus_g_method_return`.

Funkce `send_error` vytvoří novou chybu proměnnou typu `GError` zavoláním `g_error_new`. Proměnná obsahuje číslo a text vzniklé chyby. Klientská aplikace, v tomto případě grafické uživatelské rozhraní, obdrží odpověď s touto chybou. Toho docílíme zavoláním `dbus_g_method_return_error`.

Při práci se sběrnici D-Bus je využito vygenerovaného souboru `dbus_service_glue.h`. Tento soubor byl vygenerován ze souboru `org.fedoraproject.SimpleFileShare.xml` příkazem:  
`dbus-binding-tool --prefix=daemon --mode=glib-server \`  
`../data/org.fedoraproject.SimpleFileShare.xml > dbus_service_glue.h`

Soubor `org.fedoraproject.SimpleFileShare.xml` má formát xml souboru. Obsahuje navržené rozhraní `org.fedoraproject.SimpleFileShare` a definuje objekt `/org/fedoraproject/SimpleFileShare`, který je registrován na systémové sběrnici D-Bus. Navržené rozhraní má tři metody. První metoda se jménem `get_share_status` má dva parametry (jeden vstupní a jeden výstupní). Další metody mají pouze jeden vstupní parametr. Tyto metody jsou pojmenovány `setup_share` a `delete_share`.

Obsah souboru `org.fedoraproject.SimpleFileShare.xml`:

```
<!DOCTYPE node PUBLIC
"-//freedesktop//DTD D-BUS Object Introspection 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/introspect.dtd" >

<node name="/org/fedoraproject/SimpleFileShare">
  <interface name="org.fedoraproject.SimpleFileShare">
    <method name="get_share_status">
      <arg type="s" name="path" direction="in" />
      <arg type="as" name="status" direction="out" />
    </method> <method name="setup_share">
      <annotation name="org.freedesktop.DBus.GLib.Async" value="" />
      <arg type="as" name="parameters" direction="in" />
    </method>
```

Pokračování obsahu souboru `org.fedoraproject.SimpleFileShare.xml`:

```
<method name="delete_share">
    <annotation name="org.freedesktop.DBus.GLib.Async" value=""/>
    <arg type="s" name="path" direction="in" />
</method>
</interface>
</node>
```

Simple File Share démon je spuštěn až při prvním požadavku jedné z metod. Toho docílíme vytvořením D-Bus service souboru. Tento soubor je nakopírován do adresáře `/usr/share/dbus-1/system-services/`. Parametr `exec` určuje program, který se spustí při pokusu o připojení k rozhraní, jehož název je zadán v parametru `name`. Poslední položkou je parametr `user`, definující uživatelský účet, pod kterým se má aplikace spustit. V tomto případě účet administrátora systému.

Obsah D-Bus service souboru `org.fedoraproject.SimpleFileShare.service`:

```
[D-BUS Service]
Name=org.fedoraproject.SimpleFileShare
Exec=/usr/sbin/sfshared
User=root
```

Další omezení je popsáno v souboru `org.fedoraproject.SimpleFileShare.conf` uloženého v adresáři `/etc/dbus-1/system.d/`. Tento konfigurační soubor má opět podobu xml souboru a je zde určen vlastník rozhraní `org.fedoraproject.SimpleFileShare`. Nikdo jiný než uživatel `root` si nemůže rozhraní zaregistrovat. Přístup (volání metod) k tomuto rozhraní není omezen.

Obsah souboru `org.fedoraproject.SimpleFileShare.conf`:

```
<!DOCTYPE busconfig PUBLIC
"-//freedesktop//DTD D-BUS Bus Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>
    <policy user="root">
        <allow own="org.fedoraproject.SimpleFileShare"/>
    </policy>
    <policy context="default">
        <allow send_destination="org.fedoraproject.SimpleFileShare"/>
    </policy>
</busconfig>
```

System PolicyKit musí mít nadefinovaná pravidla, kterými se bude řídit. Démon registruje dvě metody na systémové sběrnici D-Bus, které mohou být provedeny pouze pokud má volající aplikace práva administrátora systému. Identifikátory metod společně s pravidly jsou popsány v souboru `org.fedoraproject.SimpleFileShare.policy` uloženém v adresáři akcí systému PolicyKit `/usr/share/polkit-1/actions/`. Metody jsou jednoznačně určeny nastavením parametru `action id`. Pro tento případ to jsou metody `setup_share` a `delete_share`. U obou je hodnotou `auth_admin_keep` nastaveno vyžádání oprávnění administrátora.

Obsah souboru `org.fedoraproject.SimpleFileShare.policy`:

```
<!DOCTYPE policyconfig PUBLIC
"-//freedesktop//DTD PolicyKit Policy Configuration 1.0//EN"
"http://www.freedesktop.org/standards/PolicyKit/1.0/policyconfig.dtd">
<policyconfig>
<vendor>SimpleFileShare</vendor> <vendor_url>
    git://fedorapeople.org/home/fedora/lipoja/public_git/sfshare.git
</vendor_url>
    <action id="org.fedoraproject.SimpleFileShare.setup_share">
        <description>Setup share</description>
        <message>Writing to smb.conf requires authentication</message>
        <defaults>
            <allow_any>auth_admin_keep</allow_any>
            <allow_inactive>auth_admin_keep</allow_inactive>
            <allow_active>auth_admin_keep</allow_active>
        </defaults>
    </action>

    <action id="org.fedoraproject.SimpleFileShare.delete_share">
        <description>Delete share from smb.conf</description>
        <message>Writing to smb.conf requires authentication</message>
        <defaults>
            <allow_any>auth_admin_keep</allow_any>
            <allow_inactive>auth_admin_keep</allow_inactive>
            <allow_active>auth_admin_keep</allow_active>
        </defaults>
    </action>
</policyconfig>
```



## 5.3.2 Funkce pracující s konfiguračním souborem `smb.conf`

Veškeré funkce, které pracují přímo s konfiguračním souborem systému Samba, jsou implementovány v souboru `samba_share.c`. Pro přehlednost je zde implementována datová struktura typu `TSmbConfItem`. Tato struktura uchovává hodnoty základních parametrů jedné položky neboli sekce. Podle potřeby je pak vytvářeno pole prvků typu `TSmbConfItem`. Pro vytvoření proměnné typu `TSmbConfItem` a současné vyplnění hodnot lze použít funkce `smbconf_item_new` se zadanými parametry, které obsahují hodnoty základních parametrů. Chceme-li pouze připravit proměnnou bez zapsání hodnot, můžeme zavolat `smbconf_item_new0`. Proměnná nebo prvek pole tohoto typu musí být později uvolněn z paměti funkcí `smbconf_item_free`. Pro práci s polem prvků typu `TSmbConfItem` využijeme funkci `shared_items_array_new` pro jeho vytvoření a `shared_items_array_free` pro odstranění.

## 5.3.3 Pomocné funkce

Důležitou funkcí implementovanou v souboru `samba_share.c` je funkce, sloužící k načtení všech sekcí popisující sdílení (vyjma speciální sekce). Funkce má identifikátor `load_smb_conf` a musí jí být předán jeden parametr v podobě ukazatele na pole prvků. Návrátová hodnota je při úspěšném průchodu rovna nule. Vyskytne-li se chyba, je návratová hodnota nastavena na číslo vzniklé chyby. Na začátku těla této funkce se zjišťuje, zda existuje konfigurační soubor určený cestou, která je uložena v proměnné `smb_conf_path`. Poté je soubor otevřen pro čtení a jsou z něj načítány jednotlivé řádky. Algoritmus popisuje obrázek 5.5.

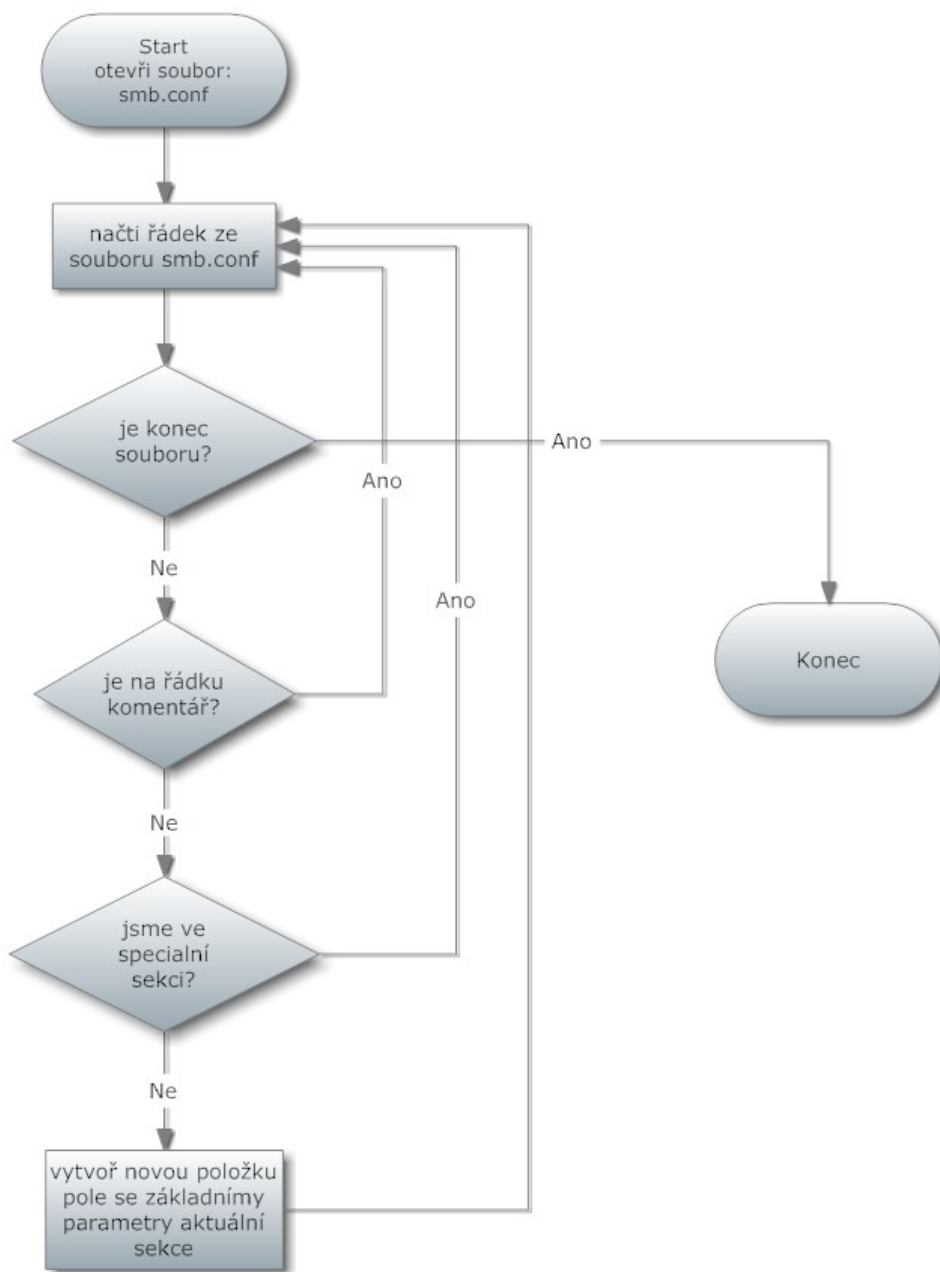
K rozpoznání parametrů sekce a jejich hodnot byla napsána funkce `parse_to_share_item`. Vstupní parametr je aktuálně načtený řádek konfiguračního souboru. Tento řádek je analyzován. Pokud bylo nalezeno jedno z klíčových slov definovaných v poli `keywords`, je hodnota nalezeného parametru nastavena i ve výstupním parametru funkce `parse_to_share_item`, kterým je ukazatel na prvek pole typu `TSmbConfItem`.

V souboru `samba_share.c` jsou popsány i dvě podobné sobě funkce. Je to funkce `is_shared_item` resp. funkce `share_name_exist`. Obě mají jako vstupní parametr ukazatel na pole sdílených sekcí načtených z konfiguračního souboru. Liší se pouze druhým parametrem. Ten udává cestu k uživatelem zvolenému adresáři resp. název sdílené sekce. Pokud je nalezena shoda, je v případě funkce `is_shared_item` vrácen ukazatel na prvek pole, který má hodnotu parametru `path` roven zadané cestě. U funkce `share_name_exist` je při nalezení stejného názvu sdílené sekce

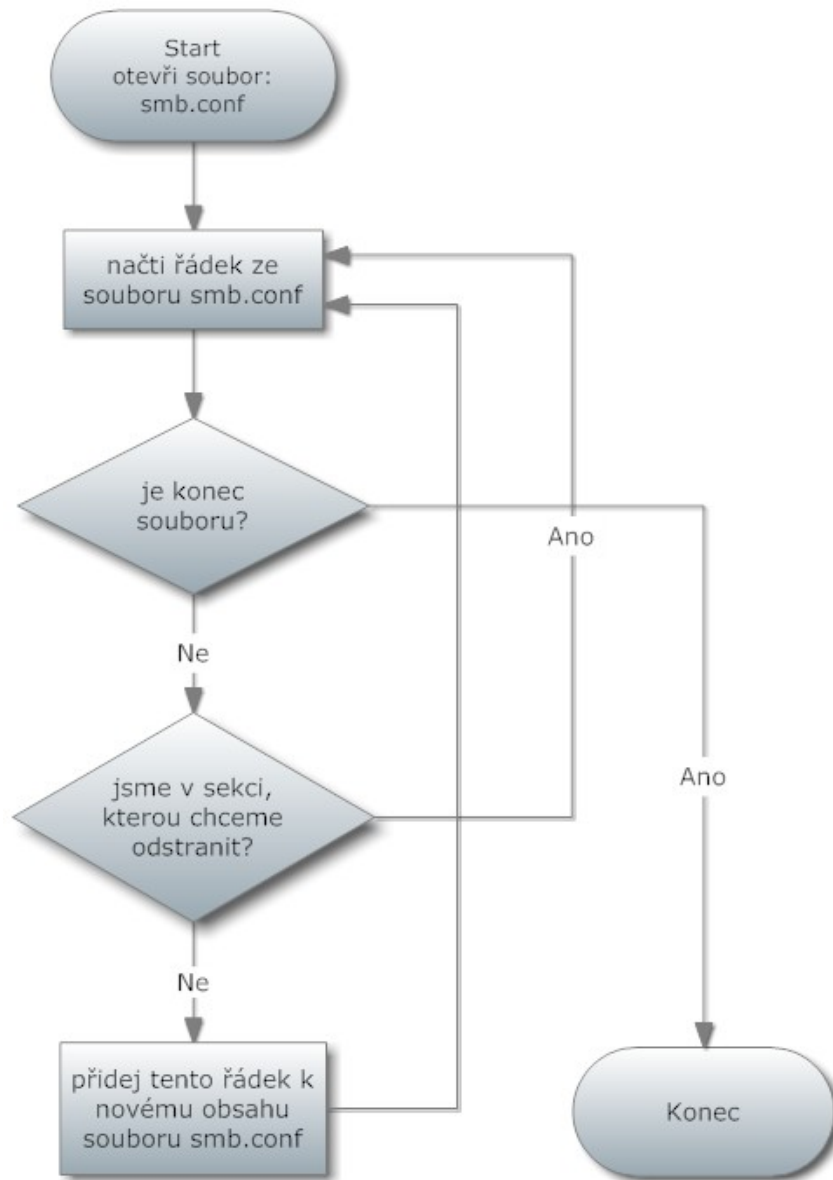
návratová hodnota rovna TRUE. Pokud bylo prohledáno celé pole a nebyla zjištěna žádná shoda názvů sekcí, je návratová hodnota rovna FALSE.

Jediná funkce, která mění obsah konfiguračního souboru `smb.conf`, je funkce `write_smbconf`. Vstupním parametrem je nový obsah konfiguračního souboru, kterým má být současný obsah přepsán. Nejdříve je otestováno, jestli konfigurační soubor existuje. Proběhne-li test v pořádku, je soubor otevřen k zápisu a zapsán celý obsah parametru funkce do souboru. Návratová hodnota je při úspěchu rovna nule nebo hodnotě chybového kódu v případě chyby.

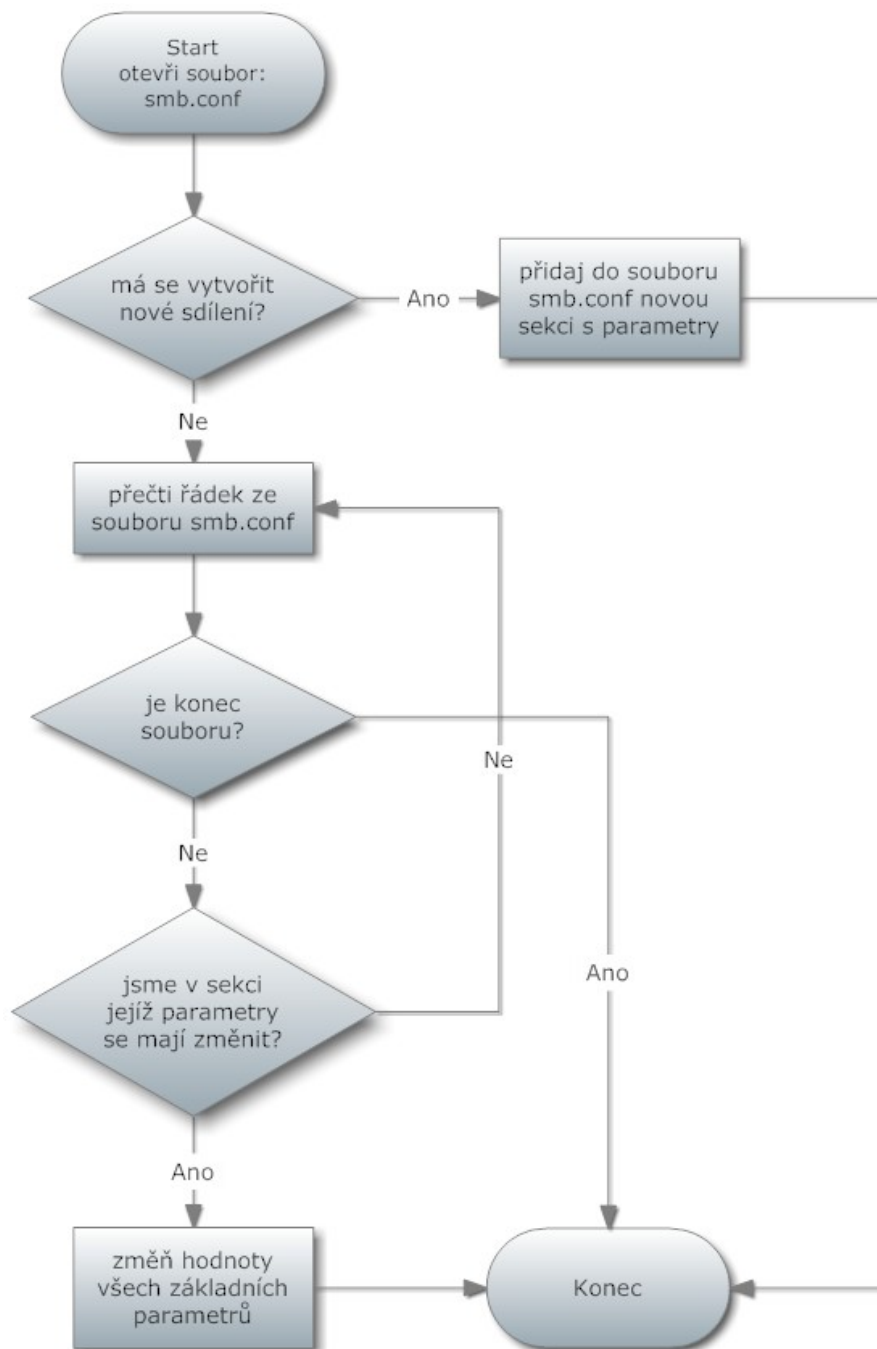
Text zapisovaný do konfiguračního souboru `Samby` je upravován ve funkci `write_share` resp. `delete_share`. Funkce `delete_share` odstraní existující sekci definující sdílení adresáře zadaného druhým parametrem. Algoritmus odstranění sekce popisuje obrázek 5.6. Funkce `write_share` je volána v případě, že je potřeba změnit hodnoty parametrů nebo vytvořit zcela novou sekci. Implementovaný algoritmus této funkce je zjednodušeněji zobrazen na obrázku 5.7.



Obrázek 5.5: Zjednodušený algoritmus funkce `load_smb_conf`.



Obrázek 5.6: Zjednodušený algoritmus funkce delete\_share



Obrázek 5.7: Zjednodušený algoritmus funkce write\_share.

### 5.3.4 Funkce implementující obsluhu D-Bus metod

K usnadnění používání funkcí popsaných v předcházející podkapitole, byly napsány funkce:

- `smb_get_share_status`,
- `smb_set_share`,
- `smb_delete_share`.

Funkce `smb_get_share_status` nejprve vytvoří pole prvků typu `TSmbConfItem`. Do tohoto pole jsou načteny všechny sekce. To zajistí funkce `load_smb_conf`. Po načtení jsou všechny prvky pole, přesněji proměnná s hodnotou parametru `path`, porovnávány se zadanou cestou. Dojde-li k nalezení sekce popisující sdílení adresáře, jsou funkcí `item_to_strv` všechny hodnoty parametrů poskládány do správného pořadí a vráceny ve druhém parametru funkce `smb_get_share_status`. Tento parametr je nastaven na hodnotu `NULL` v případě, že nebyla sekce nalezena.

Požadavek na změnu resp. vytvoření nové sekce volá funkci `smb_set_share`, které předá pole hodnot základních parametrů. Opět se načte obsah konfiguračního souboru a všechny sdílené sekce se uloží do pole. Pole spolu s proměnnou typu `TSmbConfItem`, jejíž obsah byl inicializován hodnotami ze vstupního parametru, se předá na vstup funkce `write_share`. Po úspěšném zavolání funkce `write_share` je v konfiguračním souboru `smb.conf` vytvořena nebo změněna sekce, jejíž hodnoty základních parametrů definoval uživatel.

Poslední funkcí obsluhující jednu z metod je `smb_delete_share`. Při zavolání jsou nejdříve načteny sdílené sekce do pole pomocí `load_smb_conf`. Následuje zavolání pomocné funkce `delete_share`, která zajistí odstranění sdílení z konfiguračního souborového Samby.

## 6 Závěr

Tato práce se zabývala problematikou návržení a implementování jednoduchého sdílení pomocí SMB protokolu ve správci Nautilus. Byly zde vysvětleny a popsány použité technologie i aplikace.

Při návrhu byl projekt rozdělen na tři hlavní části. Doplněk souborového správce Nautilus je oddělen od grafického uživatelského rozhraní i od Simple File Share démona. Díky tomu je přehledný a snadno upravitelný.

Jednoduché grafické uživatelské rozhraní programu Simple File Share bylo navrženo a implementováno jako samostatná aplikace. Důvodem oddělení uživatelského rozhraní od doplňku souborového správce bylo možné využití aplikace jako samostatného programu nezávislého na správci Nautilus.

Simple File Share démon byl naprogramován podle návrhu, který byl vypracován dle zadání. Démon zprostředkovává změnu údajů v konfiguračním souboru systému Samba a slouží jako propojení mezi oprávněným a neoprávněným sezením.

Projekt je vystaven v git repositáři hostovaném na serveru fedorapeople.org. Zdrojové kódy lze získat z: `git://fedorapeople.org/home/fedora/lipoja/public_git/sfshare.git`.

Program Simple File Share bude využívat firma Red Hat pro rozšíření souborového správce Nautilus. Uživatelům tak bude umožněna snadná konfigurace sdíleného adresáře a nastavení jeho základních parametrů. Tento program budou moci využít všechny linuxové distribuce, které používají desktopové prostředí GNOME a mají nainstalovaný program Nautilus.

Do budoucna by bylo dobré projekt rozšířit o možnost sdílení souborů pomocí jiného protokolu než je protokol SMB. Například sdílení protokolem WebDAV. Jednoduchou úpravou Simple File Share démona lze přidat nový modul pracující s nastavením sdílení WebDAV.

Dalším možným rozšířením je umožnit uživateli nastavení více parametrů sdílené sekce. Mezi ně patří například: povolení přístupu pouze vybraným uživatelům, nastavení masky pro vytváření souboru a adresáře. Díky nezávislosti grafického uživatelského rozhraní na doplňku správce Nautilus je možné rozšířit program i do jiných souborových manažerů využitím jejich zásuvných modulů. Těmito manažery mohou být například: Gnome-Commander nebo Tux Commander.

# Literatura

- [1] GNOME Documentation Library [online]. [cit. 2010-05-03]. File Manager.  
<http://library.gnome.org/devel/platform-overview/stable/nautilus.html>
  
- [2] GNOME Documentation Library [online]. [cit. 2010-05-03]. Introduction to GVFS.  
<http://library.gnome.org/devel/gnome-vfs-2.0/2.24/about.html>
  
- [3] GNOME Documentation Library [online]. [cit. 2010-05-05]. Extending Nautilus.  
<http://library.gnome.org/users/user-guide/2.30/gosnautilus-440.html>
  
- [4] GNOME Documentation Library [online]. [cit. 2010-05-05]. Nautilus Extension Reference Manual. <http://library.gnome.org/devel/libnautilus-extension/stable/>
  
- [5] GNOME Live! [online]. 2009-04-17 [cit. 2010-05-04]. Nautilus/Development/Extensions.  
<http://live.gnome.org/Nautilus/Development/Extensions>
  
- [6] Wikipedia, the free encyclopedia [online]. [cit. 2010-05-05]. Samba (software).  
[http://en.wikipedia.org/wiki/Samba\\_\(software\)](http://en.wikipedia.org/wiki/Samba_(software))
  
- [7] Samba documentation [online]. [cit. 2010-05-05]. smb.conf.  
<http://www.samba.org/samba/docs/man/manpages-3/smb.conf.5.html>
  
- [8] Samba documentation [online]. [cit. 2010-05-04]. Chapter 6. The Samba Configuration File.  
[http://www.samba.org/samba/docs/using\\_samba/ch06.html](http://www.samba.org/samba/docs/using_samba/ch06.html)
  
- [9] Wikipedie, otevřená encyklopedie [online]. Editována 20. 1. 2010 [cit. 2010-05-06]. D-Bus.  
<http://cs.wikipedia.org/wiki/D-Bus>
  
- [10] PENNINGTON, Havoc, et al. Software/dbus [online]. 0.5.0. [cit. 2010-05-15]. D-Bus Tutorial.  
<http://dbus.freedesktop.org/doc/dbus-tutorial.html>
  
- [11] Software/PolicyKit [online]. [cit. 2010-05-15]. PolicyKit.  
<http://www.freedesktop.org/wiki/Software/PolicyKit>



[12] PolicyKit Library Reference Manual [online]. [cit. 2010-05-15]. Theory of Operation.  
<http://hal.freedesktop.org/docs/PolicyKit/model-theory-of-operation.html>

# Seznam příloh

Příloha 1. Popis instalace programu Simple File Share

Příloha 2. Ukázka sdílení složky pomocí Simple File Share

Příloha 3. CD

# Příloha 1: Popis instalace programu Simple File Share

V této příloze bude popsána instalace programu Simple File Share. Při instalaci operačního systému Fedora 12 firmy Red Hat bylo ponecháno výchozí nastavení všech parametrů. V takto nově nainstalovaném operačním systému nelze zkompileovat program Simple File Share při prvním spuštění, protože je nezbytné doinstalovat potřebné balíčky.

Před vlastní instalací programu je dobré nainstalovat balíček Samba, který není při výchozím nastavení nainstalován. Toho docílíme příkazem: `yum -y install samba`

Ke zkompileování zdrojových souborů je nutné mít nainstalovaný překladač GCC. K tomu postačí příkaz: `yum -y install gcc`

Program Simple File Share potřebuje ke svému překladu tyto knihovny a vývojové balíčky:

- `libnautilus-extension`,
- `glib-2.0`,
- `gtk+-2.0`,
- `dbus-glib-1`,
- `polkit-gtk-1`.

Opět příkazem `yum` doinstalujeme všechny chybějící balíčky:

```
yum -y install nautilus-devel glib2-devel gtk2-devel dbus-glib-devel \
polkit-gnome-devel
```

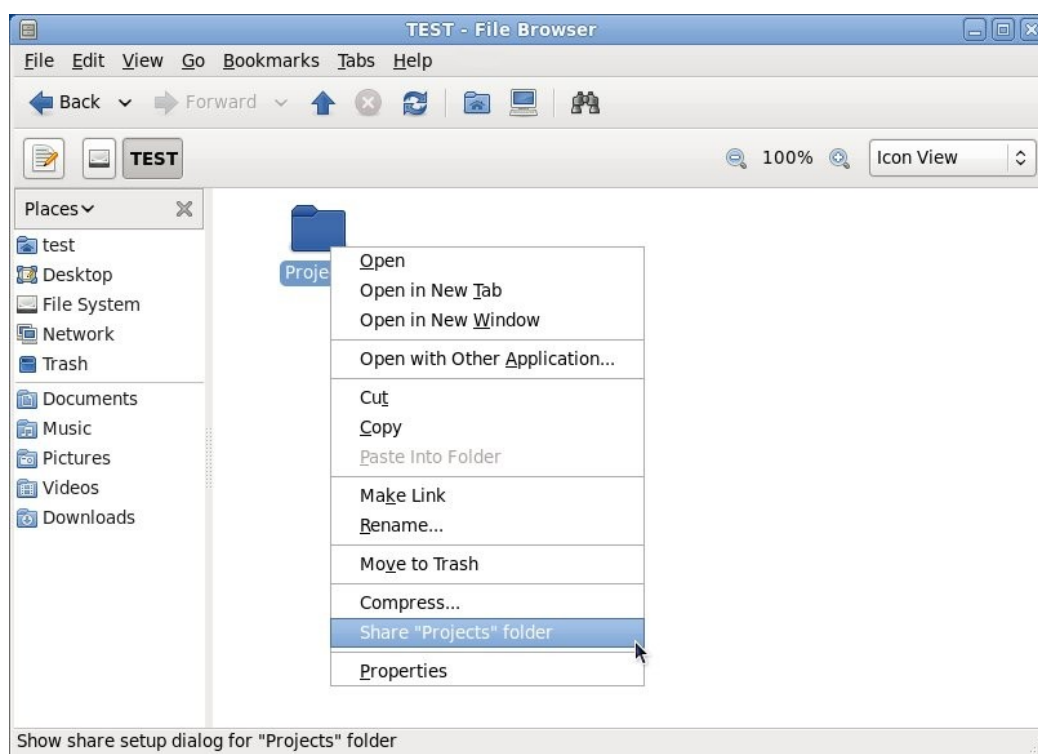
Nyní operační systém splňuje všechny požadavky, a tak můžeme přejít k vlastní instalaci programu Simple File Share. K instalaci postačí spustit `SimpleFileShare/Makefile` příkazem `make`. V této části se zkompilují všechny soubory programu. Pro instalaci je nutné mít práva administrátora systému. Ty získáme příkazem `su root` a zadáním hesla. Aplikaci nainstalujeme zadáním příkazu `make install`. Po úspěšné instalaci můžeme začít program používat. Popis, jak pracovat s programem, je v Příloze 2.

# Příloha 2: Ukázka sdílení složky pomocí

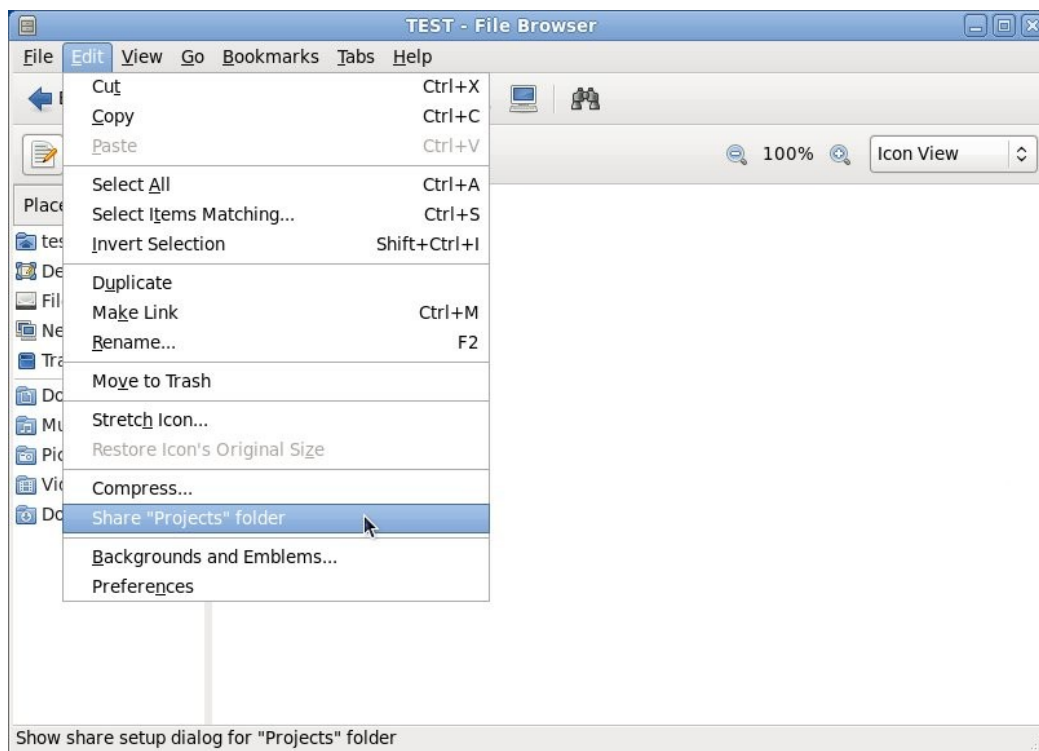
## Simple File Share

Bylo vytvořeno jednoduché rozhraní, ve kterém se uživatel rychle zorientuje, a proto je ovládání programu Simple File Share velice intuitivní a snadné. V této příloze bude popsán detailní postup sdílení složky pomocí doplňku správce Nautilus.

Uživatel spustí souborový manažer Nautilus. Najde adresář, který chce sdílet. Pro tuto ukázkou byl vytvořen adresář: /TEST/Projects. Uživatel zobrazí kontextové menu vybraného adresáře (obrázek P2.1) a vybere položku „Share “Projects“ folder“. Alternativní možností je kliknutí na stejnojmennou položku v menu Edit (obrázek P2.2).



Obrázek P2.1: Kontextové menu adresáře Projects



Obrázek P2.2: Položka sdílení v nabídce Úpravy

Pokud uživatel vybere volbu sdílení adresáře doplněk správce Nautilus spustí grafické uživatelské rozhraní programu Simple File Share (obrázek P2.3). Tímto je uživatel vyzván k vyplnění údajů o sdílení.



Obrázek P2.3: GUI programu Simple File Share

Nejdříve si uživatel vybere, zda chce adresář sdílet volbou Share with. Zde zvolí „Do not share this folder“ pokud adresář nechce sdílet a nebo „Windows share [Samba]“, chce-li nastavit sdílení vybraného adresáře. Povinným údajem je Share name, udávající jméno sdílené složky, které bude zobrazeno při připojení vzdáleného uživatele. Zaškrtnutím volby Read only access bude

vzdáleným uživatelům odebráno oprávnění zápisu do této složky. Pokud má být složka zpřístupněna uživatelům, kteří nemají nastavený vlastní účet, lze zatrhnout položku Allow GUEST access, která jim povolí přístup do složky. Comment je posledním a nepovinným políčkem v grafickém uživatelském rozhraní, které má pouze informativní charakter. Je-li formulář vyplněn, mohou se potvrdit změny stisknutím tlačítka Save (obrázek P2.4).



Obrázek P2.4: Vyplněné údaje definující vlastnosti sdílení

Nemá-li uživatel dostatečná oprávnění ke změně konfiguračního souboru `smb.conf`, je požádán systémem PolicyKit o autentizaci (obrázek P2.5).



Obrázek P2.5: Výzva k autentizaci systémem PolicyKit

Po úspěšné autentizaci jsou všechny údaje poslány přes systémovou sběrnici D-Bus Simple File Share démonu, který obslouží požadavek. V tomto případě se jednalo o vytvoření nového sdílení. Démon vytvořil novou sekci v konfiguračním souboru systému Samba a nastavil hodnoty základních parametrů, které definoval uživatel v grafickém uživatelském rozhraní Simple File Share. O bezchybné provedení je uživatel informován zprávou zobrazenou na obrázku P2.6.



Obrázek P2.6: Úspěšné vytvoření nového sdílení