

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SYNCHRONIZACE ADRES ELEKTRONICKÉ POŠTY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETER MIKUŠ

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SYNCHRONIZACE ADRES ELEKTRONICKÉ POŠTY

SYNCHRONIZATION OF EMAIL ADDRESSES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETER MIKUŠ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR MATOUŠEK, Ph.D.

BRNO 2008

Zadání bakalářské práce

Řešitel: **Mikuš Peter**

Obor: Informační technologie

Téma: **Synchronizace adres elektronické pošty**

Kategorie: Počítačové sítě

Pokyny:

1. Prostudujte způsoby výběru a ukládání emailových adres poštovními programy včetně adresářových služeb.
2. Prostudujte podobné systémy na synchronizaci souborů (rsync) či HTML bookmarks (foxmarks synchronizer).
3. Prozkoumejte možnosti ukládání emailových adres z poštovních klientů na centrálním serveru a následné čtení/synchronizace těchto údajů klienty.
4. Navrhněte model systému pro ukládání emailových adres např. pomocí LDAP serveru. V modelu uvažujte také způsob zabezpečení přenosu i uložených dat.
5. Vytvořte funkční implementaci navrženého systému pro vybraný OS a poštovního klienta.
6. Zhodnoťte výsledky své práce a možné rozšíření projektu.

Literatura:

- RFC 2251 - Lightweight Directory Access Protocol (v3), dokument dostupný na <http://www.faqs.org/rfcs/rfc2251.html>
- T.A.Howes etc.: Understanding and Deploying LDAP Directory Services, Addison Wesley, 2002
- RFC 2822 - Internet message format, dokument dostupný na <http://www.faqs.org/rfcs/rfc2822.html>

Při obhajobě semestrální části projektu je požadováno:

- body 1-4

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

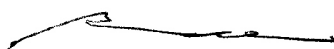
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Matoušek Petr, Ing., Ph.D.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2007

Datum odevzdání: 14. května 2008

L.S.



doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Peter Mikuš**
Id studenta: 78858
Bytem: Tatranská 6313/23, 974 11 Sásová
Narozen: 21. 10. 1985, Banská Bystrica
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Synchronizace adres elektronické pošty
Vedoucí/školitel VŠKP: Matoušek Petr, Ing., Ph.D.
Ústav: Ústav informačních systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

..... 

Autor

Abstrakt

Synchronizácia predstavuje proces udržiavania konzistencie dát uložených na viacerých miestach. Nástrojov určených k synchronizovaniu dát existuje celá rada. Od týchto aplikácií očakávame, aby synchronizácia prebehla rýchlo, spoľahlivo a bezpečne. V tejto práci sa zaoberám spôsobmi ukladania emailových adries poštovými programami a dostupnými možnosťami synchronizácie súborov. Na základe získaných poznatkov navrhmem a implementujem funkčný model. Cieľom a motiváciou práce je vytvoriť vlastné riešenie, ktoré by plnilo svoju úlohu v oblasti synchronizácie emailových adries.

Klíčová slova

synchronizácia, emailové adresy, adresárove služby, dáta, rsync, Foxmarks

Abstract

Synchronization represents the process of keeping consistency of data stored in one or more locations. There are several applications with this function available on the market. We expect from applications to synchronize data quickly, reliable and safely. In my work I am concerned with the ways of storing email addresses by mail applications and available resources of file synchronization. As the result I will make and implement functional model. The purpose and motivation of this work is to create own solution, which will perform the synchronization of email addresses.

Keywords

synchronization, email addresses, directory services, data, rsync, Foxmarks

Citace

Peter Mikuš: Synchronizace adres elektronické pošty, bakalářská práce, Brno, FIT VUT v Brně, 2008

Synchronizace adres elektronické pošty

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Petra Matouška Ph.D.

.....

Peter Mikuš
7. května 2008

© Peter Mikuš, 2008.

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

1	Úvod	2
2	Synchronizácia adries	3
2.1	Práca s adresami	3
2.1.1	Formát LDIF	4
2.1.2	Formát vCard	5
2.1.3	Formát CSV	5
2.1.4	Mozilla Address Book	6
2.2	Adresárové služby	7
2.3	Dostupné nástroje	8
2.3.1	RSync	9
2.3.2	Foxmarks	9
2.3.3	OpenSync	10
2.3.4	SyncML	10
2.3.5	Plaxo	12
2.4	Možnosti synchronizácie	13
3	Vlastné riešenie	15
3.1	Návrh systému	15
3.2	Implementácia	15
3.2.1	Klient	15
3.2.2	Štruktúra rozšírenia	15
3.2.3	GUI	16
3.2.4	Ovládanie a inštalácia	17
3.2.5	Synchronizácia	18
3.2.6	Server	19
3.3	Mozilla API	20
3.3.1	XPCOM	21
3.3.2	XML User Interface Language	21
3.3.3	XULRunner	21
3.3.4	XPIInstall	21
3.3.5	Vývojové nástroje	21
3.4	Bezpečnosť	22
3.5	Testovanie	22
3.6	Možnosti rozšíriteľnosti	22
4	Záver	24

Kapitola 1

Úvod

Počítače prenikli do bežného života a ich použitie sa stáva samozrejmosťou. Významnú časť pri práci s počítačom tvoria dáta. Dáta majú najrôznejšiu formu a obsah. Okrem dokumentov či multimediálneho obsahu, môžu zaznamenávať komunikáciu, predstavovať informácie o organizácii času alebo osobách, s ktorými komunikujeme.

V prípade, že chceme mať dáta dostupné z viacerých miest, vzniká potreba tieto dáta synchronizovať. Problém synchronizácie je jeden z najznámejších problémov spojených s dátami uloženými na viacerých miestach. Na synchronizáciu sa možno pozeráť ako na istú formu zálohovania. Synchronizácia by mala prebiehať rýchlo, spoľahlivo a pokiaľ možno bezpečne.

Cieľom tejto práce je zaoberať sa spôsobmi ukladania emailových adries poštovými programami, možnosťami ukladania adries na centrálnom servery ako aj dostupnými možnosťami synchronizácie súborov. Na základe získaných poznatkov navrhнем model systému a zamyslím sa nad otázkami jeho bezpečnosti. V praktickej časti implementujem navrhnutý systém pre vybraného poštového klienta a operačný systém.

V druhej kapitole sa budem venovať teoretickému pohľadu na prácu s emailovými adresami. Budú spomenuté jednotlivé formáty súborov určené na ukladanie. Ukladanie bude demonštrované na vybraných poštových programoch. Uvediem možnosti synchronizácie a podobne zamerané riešenia a služby. V tretej kapitole popíšem vlastný model a jeho implementáciu. Zhodnotím jeho vlastnosti a zamyslím sa nad otázkami bezpečnosti a možnosťami rozšírenia tohto modelu. Stručne popíšem rozhranie vytvoreného programu, jeho implementáciu a základnú prácu s ním. V záverečnej kapitole zhodnotím svoju prácu a dosiahnuté výsledky.

Kapitola 2

Synchronizácia adries

V následujúcej kapitole uvediem teoretický úvod do problematiky synchronizácie a popíšem dostupné aplikácie. Spomením jednotlivé spôsoby a formáty určené pre ukladanie a prácu s emailovými adresami.

Synchronizácia dát je proces udržiavania konzistencie a harmónie medzi dátami vzdialených zdrojov v čase [9].

Synchronizáciou rozumieme automatické prenášanie zmien na množine dát medzi dvoma alebo viacerými zariadeniami. Napríklad, adresár z jedného zariadenia môže byť synchronizovaný s adresárom nachádzajúcim sa na inom zariadení. Pod zariadením rozumieme napríklad počítač, server alebo prenosné zariadenie (mobilný telefón, PDA, hudobný prehrávač, ...). Tieto zariadenia majú rôzne využitie od čoho závisí aj množina dát určená k synchronizácií. Synchronizovať možno adresáre, súbory, kalendáre, kontakty alebo nastavenia programov.

2.1 Práca s adresami

Jedna z najzákladnejších funkcií, ktorú očakávame od emailového klienta, je ukladanie kontaktných informácií o osobách, s ktorými často komunikujeme. Podpora ukladania kontaktov lokálne na strane aplikácie, tzv. „adresár“ je prítomná takmer vo všetkých súčasných emailových aplikáciach.

Adresár môže byť používaný špeciálne konkrétnou aplikáciou (napr. Mozilla Thunderbird) alebo môže byť zdieľaný pre viacero aplikácií (napr. Microsoft Windows Mail). Jednotlivé aplikácie sa vyznačujú vlastnou implementáciou a prístupom k adresáru. Okrem emailovej adresy je možné mať pri kontakte priradené ďalšie kontaktné údaje. Typicky je umožnené priradiť napr. adresu, fax, adresu webovej prezentácie, telefónne číslo, a iné. Kategórie a položky sa vyskytujú vo viacerých inštanciách (súkromné, zamestnanie, ...). Základné typy súborov, s ktorými aplikácie dokážu pracovať sú LDIF, CSV, vcf, txt. V tabuľke 2.1 je uvedený prehľad najpoužívanejších klientskych a podporovaných formátov pri práci s adresami.

Zatiaľ čo formáty LDIF, csv, vcf a tab sú popísané príslušným štandardom, *wab* (Windows Address Book) je proprietárnym formátom spoločnosti Microsoft. Je využívaný viacerými programami napr. Outlook, Windows Mail, Outlook express. Formát *vcf* sa od ostatných odlišuje tým, že jeden súbor obsahuje práve jeden kontakt.

Klient	Podpora formátov (Import/Export)						LDAP
	LDIF	vcf	txt	csv	wab	tab	
Mozilla Thunderbird	●/●	○/○	○/●	●/●	○/○	●/●	●
Windows Mail	●/○	●/●	○/○	●/●	●/○	○/○	●
Evolution	●/○	●/●	○/○	●/○	○/○	○/○	●
Outlook 2007	●/○	●/●	○/○	●/●	●/○	○/○	●
○ - nepodporuje, ● - podporuje							

Tabulka 2.1: Prehľad aplikácií a nimi podporované formáty

2.1.1 Formát LDIF

LDIF (LDAP Data Interchange Format) je štandardizovaný formát súboru, ktorý slúži na reprezentáciu dát v LDAP adresároch. Je definovaný štandardom a bol navrhnutý v roku 1990 [3].

Obsah adresára vyjadruje ako množinu záznamov, jeden záznam pre každý objekt. Rovnako reprezentuje požiadavky na aktualizáciu alebo vymazanie už existujúcich záznamov poprípade na pridanie nového záznamu. Jeden záznam pre každý požiadavok. V oboch prípadoch sú dáta prezentované v textovej forme. Súbor slúži ako prostriedok pre import a export dát z adresáru. Príklad LDIF súboru (kontakt z programu Mozilla Thunderbird):

```
dn: cn=Peter Mikus,mail=xmikus06@stud.fit.vutbr.cz
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
objectclass: mozillaAbPersonAlpha
givenName: Peter
sn: Mikus
cn: Peter Mikus
mail: xmikus06@stud.fit.vutbr.cz
```

LDIF súbor sa skladá z blokov oddelených novým riadkom. Každý riadok reprezentuje jednu položku. Základná štruktúra:

```
dn: <rozlišovacie meno>
objectClass: <trieda objektu>
<atribút>: <hodnota>
```

Atribúty popisujú vlastnosti objektu. Atribút **dn** (Distinguished name) je tzv. „rozlišovacie“ meno. Jednoznačne identifikuje záznam v globálnom mennom priestore adresárového stromu. Skladá sa z *RDN* (Relative Distinguished Name) tj. relatívnych rozlišovacích mien. RDN unikátne špecifikuje záznam v rámci jednej vetvy stromu. Rozlišovacie meno sa skladá z jednotlivých relatívnych rozlišovacích mien. Postupnosť relatívnych rozlišovacích mien tvorí cestu od koreňa k záznamu.

Každému záznamu je priradená „trieda záznamu“ - `objectClass`. Trieda záznamu určuje, ktoré atribúty sa môžu a ktoré sa musia vyskytovať. Atribúty majú pridelený jednoduchý typ ako napríklad celé číslo alebo reťazec. Trieda tiež definuje operácie nad adresárom a upravuje umiestnenie záznamu v adresárovej štruktúre. Tieto pravidlá sú definované v schéme databázy. Nad objektami existuje dedičnosť.

2.1.2 Formát vCard

Formát pre elektronické vizitky vCard (VersitCard), slúži na výmenu osobných údajov. Formát bol predložený v roku 1995 a jeho súčasná verzia 3.0 je popísaná štandardom [8].

vCard môže obsahovať meno, adresu, email, obrázky, telefónne číslo. Najčastejšie sa prikladajú ako príloha do emailu existujú však aj iné spôsoby posielanie napr. WWW. Mikroformát, ktorý umožňuje vkladať vizitky priamo do internetových stránok sa nazýva hCard. Je presnou reprezentáciou vCard formátu v sémantike (X)HTML. Jedná sa o textový formát [2]. Príklad kontaktu vo formáte vCard:

```
BEGIN:VCARD
VERSION:3.0
PROFILE:VCARD
FN:Peter Mikus
N:Mikus;Peter;;;
EMAIL;TYPE=INTERNET;TYPE=PREF:xmikus06@stud.fit.vutbr.cz
UID:a834a9da937c6cbd2c511ff64e231b4f
REV:2008-01-30T14\ :51\ :55Z
END:VCARD
```

vCard môže obsahovať jeden alebo viac objektov, ktoré sa môžu zoskupovať. Objekt je kolekciou jednej alebo viacerých vlastností. Každá vlastnosť má jedinečné meno (VERSION, FN, N, EMAIL, ...). Hodnoty parametrov sú oddelené znakom bodko-čiarka (;). Bodko-čiarkou sú oddelené rovnako aj zložené hodnoty. Ak je použitá v hodnote parametra, je potrebné pred tento znak uvádzať znak spätného lomítka (\). Východzie kódovanie súboru je 7-bitové ale je možné ho zmeniť samostatným parametrom ENCODING s hodnotou BASE64, QUOTED-PRINTABLE alebo 8BIT. Objekt je jednoznačne identifikovaný pomocou UID a je ohraničený kľúčovými slovami BEGIN a END.

Kontakt je možné vytvoriť napr. online nástrojom [vCardMaker](#).

2.1.3 Formát CSV

CSV (comma-separated values) je formát súboru vhodný pre uchovanie tabuľkových dát. Jedná sa o textový súbor, ktorý sa skladá z riadkov. Hodnoty v riadku sú oddelené znakom, spravidla čiarkou (,). Je možné definovať iné (vlastné) oddeľovacie znaky napr. tabulátor (TSV tab-separated value) alebo bodko-čiarka [5]. Formát nemá žiadnu formálnu špecifikáciu. Príklad súboru CSV:

```
First Name,Last Name,Display Name,Nickname,Primary Email,Phone,
Peter,Mikuš,Peter Mikuš,xmikus06,xmikus06@stud.fit.vutbr.cz,,
Foo,Bar,Foo Bar,xfooba01,foo@bar.com,555-1234,
```

Jeden záznam predstavuje jeden riadok. V prípade, že hodnota v zázname obsahuje oddeľovací znak, musí byť táto hodnota uzatvorená v uvozovkách. Prebytočné medzery pred

a za oddeľovacím znakom môžu byť orezané. CSV formát nepotrebuje špeciálne kódovanie ani poradie bajtov. Za ukončenie riadku sa považuje znak CR\LF.

2.1.4 Mozilla Address Book

MAB (Mozilla Address Book) je formát súboru, ktorý slúži na ukladanie kontaktov v programe Mozilla Thunderbird a Netscape. Formát ma príponu `.mab` a je uložený v adresári s používateľským profilom aplikácie. Ukladá sa teda zvlášť pre každého používateľa. Každý adresár s kontaktmi predstavuje samostatný súbor uložený na disku. Mená súborov uložených na disk sa riadia konvenciami uvedenými v tabuľke 2.2.

Zobrazované meno	Meno súboru	Pozícia v zozname
Personal Address Book	abook.mab	Prvé miesto v zozname
Collected Addresses	history.mab	Druhé miesto v zozname
Prvý vytvorený adresár	abook-1.mab	Podľa dátumu vytvorenia
n-tý vytvorený adresár	abook-n.mab	Podľa dátumu vytvorenia
Prvý importovaný adresár	impab.mab	Podľa dátumu vytvorenia
n-tý importovaný adresár	impab-n.mab	Podľa dátumu vytvorenia

Tabuľka 2.2: Konvencie ukladania kontaktov programom Thunderbird

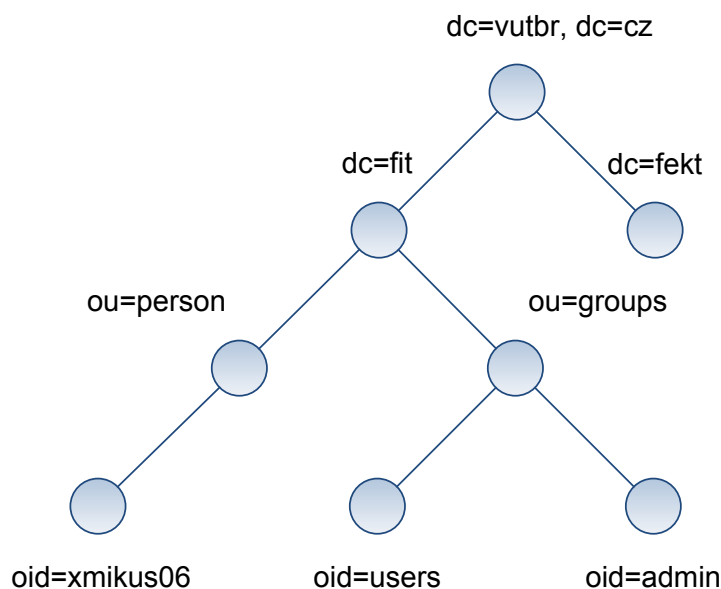
MAB je textový (plain) formát, ktorý podporuje komprimáciu. Na prvom sa nachádza identifikácia súboru `// <!-- <mdb:mork:z v="1.4"/> -->`. Za identifikáciou nasleduje výpis definovaných položiek (atribúty kontaktu) a k nim priradené kódy. Na tieto kódy sa pri jednotlivých záznamoch možno odkazovať. Tým sa dosiahne menšia veľkosť súboru. Za týmto výpisom nasleduje zoznam použitých unikátnych hodnôt, ktoré majú opäť priradené kódy. Samotný záznam tak obsahuje kombináciu kódov definovaných na začiatku súboru. Ukážka súboru MAB:

```
// Definovanie kľúčových slov
<(87=1)(81=Peter)(82=Miku$C5$A1)(83=)(84=Peter Miku$C5$A1)(85=xmikus06)
(86=xmikus06@stud.fit.vutbr.cz)>
{-1:^80 {(k^C0:c)(s=9)}
// Nasleduje záznam o osobe, kde jednotlivé kódy značia atribúty kontaktu
// a priradený kód definovaný v kľúčových slovách
[1:^82(^BF=1)]
[-1(^83^81)(^84^82)(^85=)(^86=)(^87^84)(^88^85)(^89^86)(^8A^86)(^8B=)
(^8C=)(^8D=)(^8E=0)(^8F=0)(^90=0)(^91=)(^92=)(^93=)(^94=)(^95=)
(^96=)(^97=)(^98=)(^99=)(^9A=)(^9B=)(^9C=)(^9D=)(^9E=)(^9F=)(^A0=)
(^A1=)(^A2=)(^A3=)(^A4=)(^A5=)(^A6=)(^A7=)(^A8=)(^A9=)(^AA=)(^AB=)
(^AC=)(^AD=)(^AE=)(^AF=)(^B0=)(^B1=)(^B2=)(^B3=)(^B4=)(^B5=)(^B6=)
(^B7=)(^B8=)(^B9=)(^BA=)(^BB=)(^BC=0)(^BD=1)]]
```

2.2 Adresárové služby

Pod pojmom adresárová služba rozumieme špecializované aplikácie, ktoré slúžia k ukladaniu veľkého množstva dát, ich organizácií a prístupu k nim. Plní úlohu abstraktnej vrstvy medzi užívateľom a zdieľanými zdrojmi. Definuje základné operácie vkladania, mazania, modifikácie a vyhľadávania nad dátami [1].

Pre jednoduchú predstavu možno uvažovať, že dáta sú uložené vo forme záznamov, pričom každý záznam obsahuje atribúty. Atribút je nositeľom dát (vlastností), tj. má hodnotu. Usporiadanie záznamov je štrukturované a hierarchické. Sú usporiadané hierarchicky v adresárovom strome tzv. „directory information tree - DIT“. DIT predstavuje konkrétny návrh štruktúry adresárového stromu, členenie záznamov a ich informácií do hierarchicky usporiadaných skupín. Každý záznam má unikátne meno globálne v rámci stromu a každý atribút má unikátne meno v rámci záznamu. Na obrázku 2.1 je príklad znázorňujúci adresárovú štruktúru. Vrcholy tvoria záznamy a hrany určujú vzťahy medzi nimi.



Obrázek 2.1: Príklad stromovej adresárovej štruktúry

Adresárova služba poskytuje prístupové rozhranie k dátam obsiahnutým v jednom alebo viacerých menných priestoroch. V prípade adresárových služieb sa jedná o špecializovanú databázu určenú predovšetkým pre aplikácie, ktoré intenzívne prístupujú k dátam (čítanie, prehľadávanie) ale dáta príliš často nemodifikujú. Modifikácia sa uskutočňuje jednoduchými prostriedkami (žiadne transakcie). Adresárové služby umožňujú primárne vyhľadávať objekty podľa špecifikovaných podmienok. Slúžia tiež k získavaniu informácií o konkrétnych objektoch [1].

Typické využitie adresárových služieb by sme našli napríklad pri aplikáciách ako telefónny zoznam, používateľské účty, autentifikácia používateľov alebo firemná databáza zamestnancov. V uvedených príkladoch by sme mohli zvoliť ako záznam človeka, ktorý by obsahoval atribúty napr. meno, adresu, telefónne číslo, email, heslo (v prípade používateľských

úctov), zaradenie (firemná databáza zamestnancov). Celková štruktúra závisí na konkrétnej oblasti použitia a je možné ju definovať podľa potrieb.

Príkladom implementácie adresárovej služby je štandard X.500. Je to súbor protokolov a štandardov definujúcich prácu s dátami v stromovej štruktúre. Z neho odvodený LDAP je jeho odľahčenou verziou. Implementácia nedefinuje samotné ukladanie dát ale formu ich uloženia a spôsob komunikácie. Existuje veľké množstvo implementácií adresárových služieb konkrétnych výrobcov napr. Active Directory (Microsoft), Apache Directory Server (Apache Software Foundation), Open Directory (Apple), OpenLDAP, Red Hat Directory Server (Red Hat).

Vlastnosti adresárových služieb možno zhrnúť do nasledujúcich bodov [6]:

- Špecializované prostriedky pre ukladanie dát a prístup k nim.
- Optimalizácia návrhu vzhľadom k špecifickým podmienkam, najmä:
 - Predpoklad málo sa meniacich informácií.
 - Predpoklad jednoduchých operácií s dátami.
 - Predpoklad majority prístupov, ktoré iba čítajú dáta, poprípade potrebujú dáta hľadať.

Viacero súčasných klientov dokáže pracovať s adresárovou službou, akoby formou vzdialeného adresára. Prostredníctvom dotazov je možné získať požadované informácie. Nevýhoda tohto modelu spočíva v tom, že dáta na vzdialenej službe sú dostupné iba na čítanie a nie zápis. Príkladom môže byť server `ldap.fit.vutbr.cz`.

2.3 Dostupné nástroje

Ako vyplýva z tabuľky 2.1, prácu s LDAP adresárovými službami podporujú všetky uvedené emailové aplikácie. V oblasti synchronizácie dát existuje celá rada nástrojov. Programy sa navzájom odlišujú svojim zameraním a funkcionalitou. Niektoré z nich majú špecifické využitie, iné sú určené na všeobecné použitie. Úlohou všetkých nástrojov je udržiavať rovnaké verzie súborov na viacerých miestach (typicky adresáre, zariadenia).

V praxi často požadujeme aby tieto aplikácie reflektovali zmeny v dátach pokiaľ možno rýchlo, bezpečne, a čo najspoľahlivejšie. Pokročilejšie programy poskytujú možnosť pracovať so vzdialenými adresármi po sieti alebo cez internet (`rsync`, `iFolder`). Umožňujú upozorniť na prípadné rozdiely a prenášať len zmeny, čím dosiahneme kratšieho času celého procesu. Rovnako dôležitá je bezpečnosť prenášaných dát. Program `rsync` napríklad dokáže dáta prenášať šifrovane cez SSH.

Pre účely synchronizácie mobilných zariadení existuje *SyncML* (Synchronization Markup Language). Tento jazyk ponúka nezávislý a otvorený štandard pre výrobcov mobilných zariadení. Zjednocuje a predpisuje metódy pre synchronizáciu dát (kontakty, kalendár, poznámky, ...). Medzi aplikácie podporujúce tento štandard patrí napríklad *OpenSync* alebo *TSync* (Thunderbird rozšírenie). Spoločnosť Microsoft používa pre svoje zariadenia Windows Mobile a Windows CE nástroj *ActiveSync*. Pre počítače s MacOSX je k dispozícii aplikácia *iSync*.

2.3.1 RSync

RSync je softwarová aplikácia určená k synchronizovaniu súborov a adresárov medzi dvoma miestami. Hlavnou výhodou softwaru je minimalizovanie datového prenosu použitím delta kódovania (delta kompresie).

RSync slúži ako alternatíva k *scp/rcp*. Rozdiel ale spočíva v použití algoritmu, ktorý zabezpečí rýchly prenos dát. Pri *rcp* je súbor prenášaný ako celok (pri *scp* je navyše prenos šifrovaný). Naproti tomu rSync, synchronizované súbory rozdelí na viacero menších častí a vypočíta *hash* kontrolný súčet, podľa ktorého porovná odpovedajúce časti súborov. Rozdielne časti sú potom postupne prenesené. Takto malé časti je možno navyše skomprimovať (napr. pomocou *ZLIB* kompresie).

Predpokladajme dva počítače **A** a **B**. Počítač **A** má prístup k súboru *sA* a počítač **B** má prístup k súboru *sB*, pričom súbory sú „podobné“. Počítače sú navzájom prepojené napr. prostredníctvom siete [7]. Rsync synchronizácia má nasledovný algoritmus:

1. Počítač **B** rozdelí súbor *sB* do niekoľkých navzájom neprekrývajúcich sa blokov o pevnej veľkosti *S* bajtov. Posledný blok môže byť kratší ako *S* bajtov.
2. Pre každý z týchto blokov počítač **B** vypočíta dva kontrolné súčty: slabší tzv. „rolling“, ktorý má 32 bitov a silný 128-bitový MD4 kontrolný súčet.
3. Počítač **B** pošle kontrolné súčty počítaču **A**.
4. **B** prehľadá súbor *sA* a snaží sa nájsť všetky bloky dĺžky *S* bajtov, ktoré majú rovnaký slabý aj silný kontrolný súčet ako blok *sB*.
5. Počítač **A** pošle **B** sekvenciu inštrukcií pre zostavenie kópie súboru *sA*. Každá inštrukcia je odkazom na blok súboru *sB*, alebo obsahuje nové dáta. Nové dáta sú posielané v prípade, že bloky súboru *sA* sa nenachádzajú v súbore *sB*.
6. Počítač **B** na základe inštrukcií zostaví kompletný súbor *sB* a znovu vypočíta kontrolný súčet.

Prenos môže byť realizovaný cez *ssh* protokol, čím sa dosiahne vyššieho zabezpečenia prenášaných dát. Dôležitou vlastnosťou rSync je, že na rozdiel od väčšiny iných podobných protokolov sa pri zrkadlení (*mirroring*) odohráva iba jeden prenos v každom smere. Aplikácia kopíruje súbory z/na vzdialený počítač alebo lokálne na aktuálnom počítači (nepodporuje kopírovanie medzi vzdialenými počítačmi). Dokáže bežať v dvoch režimoch. V prvom prípade je aplikácia spustená ako daemon (na pozadí, TCP port 873) a posiela súbory priamo cez TCP protokol. V druhom prípade je spustená samostatná aplikácia a na prenos sa používa *ssh* alebo *rsh* protokol.

Aplikácia je vydaná pod GNU General Public License, a domovská stránka produktu sa nachádza na adrese <http://samba.anu.edu.au/rsync/>.

2.3.2 Foxmarks

Foxmarks je rozšírenie pre internetový prehliadač Mozilla Firefox a umožňuje synchronizovať internetové záložky (*bookmarks*) z viacerých počítačov.

Rozšírenie spolieha na server, kde sú dáta uložené. Ako server možno použiť vlastný alebo využiť Foxmarks server, ktorý je voľne dostupný. Foxmarks podporuje FTP ako aj WebDAV protokol (cez HTTP alebo HTTPS).

Kedykoľvek používateľ vykoná zmenu v záložkách, aplikácia zareaguje na túto zmenu a spustí päť minútový interval. Ak je vykonaná ďalšia zmena, časovač sa resetuje. Po uplynutí intervalu, foxmarks spustí synchronizáciu prevedených zmien. V prípade, že je program ukončený skôr ako dôjde k vynulovaniu časovača, program sa opýta, či sa majú zmeny synchronizovať pred ukončením. Každú hodinu je navyše vyvolaná synchronizácia, bez ohľadu na to či bola vykonaná zmena. Jej úlohou je skontrolovať, či nedošlo k zmene z iného počítača. Automatické synchronizácie je možné vypnúť [10].

Domovská stránka rozšírenia je na adrese <http://www.foxmarks.com/>. Dostupné je aj z Mozilla repozitáru.

2.3.3 OpenSync

OpenSync je knižnica (plugin) napísaná v jazyku C a je platformne nezávislá. Je voľne dostupná pod GNU GPL licenciou.

Úlohou je synchronizovať dáta medzi počítačmi a prenosnými zariadeniami (kontakty, poznámky, úlohy, kalendáre). Podporuje viacero rôznych koncových bodov synchronizácie (telefón, PDA, PIM aplikácie, LDAP adresárové služby, ...). Má tiež podporu USB, Bluetooth, IrMC. Výhodou toho programu je, že dokáže synchronizovať aj zmeny v atribútoch rovnakého objektu, čím redukuje prípadne vzniknuté chyby a upozornenia systému. Ako knižnica, môže byť využívaná inými aplikáciami.

2.3.4 SyncML

SyncML je otvorený štandard, ktorý prináša dátovú mobilitu ustanovením jazyka pre komunikáciu medzi zariadeniami, aplikáciami a sieťami. Cieľom je umožniť pohodlnú a efektívnu synchronizáciu vzdialených dát a osobných údajov. Tento jazyk umožňuje synchronizáciu dát ako sú emaily, kontakty, kalendáre cez sieť tak, že informácie sú konzistentné, aktuálne a prístupné nezávisle na tom kde sú uložené (telefón, PDA, počítač, server). Napríklad, používateľ by mal byť schopný čítať a písať emaily na počítači alebo na mobilnom telefóne a udržiavať pritom konzistentnosť, ktoré správy boli už boli prečítané a na ktoré bolo odpovedané.

SyncML iniciatíva, zjednotená v Open Mobile Alliance (OMA), je podporovaná firmami vrátane Ericsson, Nokia, IBM, Motorola a Symbian.

Špecifikácia jazyka bola navrhnutá podporovať dva hlavné ciele [4]:

- Synchronizovať dáta na sieti z dátami v akomkoľvek mobilnom zariadení
- Synchronizovať dáta v mobilnom zariadení s dátami na sieti

Zakladné vlastnosti SyncML jazyka:

- Pracovať efektívne cez prenosové médium: Vyššia odozva na bezdrátových sieťach, obmedzená priepustnosť a nízka spoľahlivosť dát a spojenia. SyncML preto poskytuje široké možnosti synchronizačného protokolu, WAP Binary XML (WBXML) kódovanie dát a synchronizačných príkazov, pri použití jednoduchého princípu dotaz-odpoveď. Dotaz obashuje všetky zmeny a odpoveď vykoná zmeny s ohľadom na možné konflikty. Proces synchronizácie obsahuje mechanizmy pre zotavenie po chybách.
- Podporuje viacero transportných protokolov: HTTP, WAP's Wireless Session Protocol (WSP), Bluetooth's OBEX, IrDA a SMTP.

- SyncML nešpecifikuje, ako sú dáta uložené v zariadení. Popisuje len spôsob, akým sú tieto dáta prenášané cez prenosové médium. Podporované sú: vCard, email a správy (news), XML a HTML dokumenty, binárne dáta.
- Podporuje prístup k dátam z rôznych aplikácií (nezávisle na programovacom jazyku).
- Znižuje nároky na mobilné zariadenia.

SyncML štandard pre dátovú synchronizáciu pozostáva z dvoch protokolov. SyncML prezentačný protokol, ktorý definuje formát správ v XML a SyncML synchronizačný protokol, ktorý definuje akcie medzi klientom a serverom. SyncML správy sú reprezentované ako XML dokument. Musí byť presne formovaný ale nemusí byť validný. Príklad SyncML súboru:

```
<SyncML>
  <SyncHdr>
    <VerDTD>1.0</VerDTD>
    <VerProto>SyncML/1.0</VerProto>
    <SessionID>1</SessionID>
    <MsgID>1</MsgID>
    <Target><LocURI>http://www.syncml.org/sync-server</LocURI></Target>
    <Source><LocURI>IMEI:493005100592800</LocURI></Source>
    <Meta> <!--Maximálna veľkosť správy, ktorú klient dokáže prijať-->
      <MaxMsgSize xmlns='syncml:metinf'>5000</MaxMsgSize>
    </Meta>
    <Cred> <!--Authentifikácia je nepovinná-->
      <Meta><Type xmlns='syncml:metinf'>syncml:auth-basic</Type></Meta>
      <Data>QnJ1Y2UyOj9oQmVoYXZl</Data>
    </Cred>
  </SyncHdr>
  <SyncBody>
    <!-- Telo správy -->
  </SyncBody>
</SyncML>
```

Telo správy obsahuje konkrétne požiadavky ukončené značkou </final>:

```
<Get>
  <CmdID>1</CmdID>
  <Meta>
    <Type xmlns='syncml:metinf'>application/vnd.syncml-devinf+xml</Type>
  </Meta>
  <Item>
    <Target><LocURI>./devinf10</LocURI></Target>
  </Item>
</Get>

<Alert>
  <CmdID>1</CmdID>
  <Data>200</Data> <!-- 200 = TWO_WAY_ALERT -->
```

```

<Item>
  <Target><LocURI>./contacts/james_bond</LocURI></Target>
  <Source><LocURI>./dev-contacts</LocURI></Source>
  <Meta>
    <Anchor xmlns='syncml:metinf'><Last>234</Last><Next>276</Next>
  </Anchor>
</Meta>
</Item>
</Alert>

```

Proces synchronizácie začína inicializačnou fázou. Klient informuje server, ktorú databázu požaduje synchronizovať a ktorý typ synchronizácie je potrebný. Do požiadavku môže klient zahrnúť autentifikáciu (autentifikácia je nepovinná). Autentifikácia je označená entitou **Cred** v hlavičke správy (**SyncHdr**). Následne klient pošle inicializačnú správu. Databáza, ktorá bude synchronizovaná je uzavretá entitou **Alert** v tele správy (**SyncBody**). **Alert** tiež určuje typ synchronizácie [4].

Typy synchronizácie:

- **Two-way sync** - je základný typ iniciovaný klientom. Klient a server si vymenia informácie o zmenách v dátach. Server pošle nové dáta (z pohľadu klienta). Klient si aktualizuje databázu a pošle späť potrebné informácie.
- **Slow sync** - v tomto type sú všetky položky databázy klienta porovnané so všetkými položkami na strane serveru. Prebehne v prípade, keď niektorá strana stratí zoznam zmien (Change log).
- **One-way sync (klient)** - server obdrží všetky zmeny od klienta ale neposiela dáta späť.
- **Refresh sync (klient)** - klient pošle všetky dáta na server. Server prepíše svoje dáta, dátami od klienta.
- **One-way sync (server)** - klient obdrží všetky zmeny od servera ale neposiela dáta späť.
- **Refresh sync (server)** - server pošle všetky dáta klientovi. Klient prepíše svoje dáta, dátami od servera.
- **Server-alerted sync** - server upozorní klienta, že je potrebné zahájiť určitý typ synchronizácie.

2.3.5 Plaxo

Ďalšia služba určená k synchronizácií je Plaxo. Plaxo je online vzdialený adresár. Poskytuje automatickú zálohu a aktualizáciu kontaktov. Užívateľské účty ako aj dáta sú uložené na Plaxo serveroch. Pri zmene kontaktov dojde k synchronizácií a zmeny sa prejaví v ktoromkoľvek programe, ktorý sme nastavili k synchronizovaniu. Takto centrálné uložené dáta sú dostupné odkiaľkoľvek.

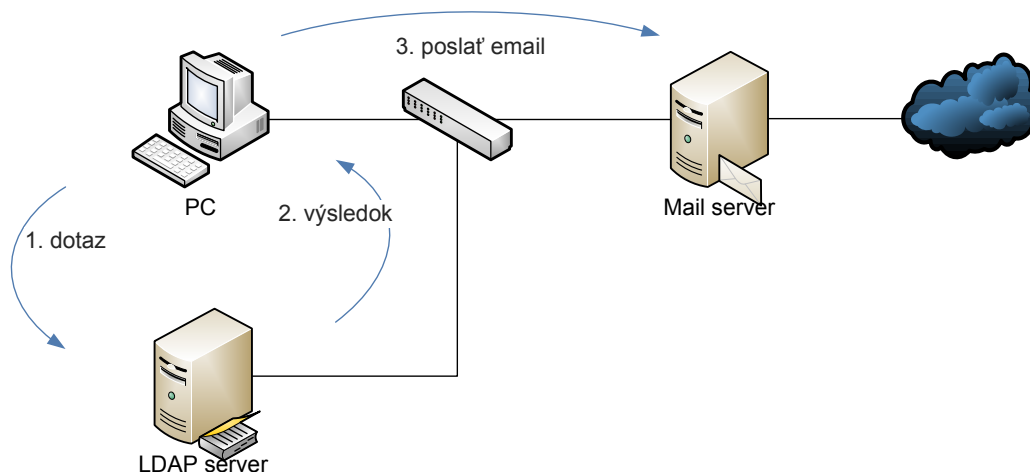
Plaxo rozšírenie je dostupné pre hlavné aplikácie vrátane Outlook/Outlook Express, Mozilla Thunderbird, a Mac OS X emailového klienta. Plaxo účet môže byť dodatočne spravovaný cez webové rozhranie z internetového prehliadača a poskytuje viacero služieb v oblasti

správy osobných údajov. Plaxo má dostupné API, ktoré možno využiť pri písaní vlastnej aplikácie. Jedná sa o proprietárne riešenie a je dostupné na adrese <http://www.plaxo.com/>.

2.4 Možnosti synchronizácie

Súčasný synchronizačný nástroj poskytuje rozdielne možnosti práce s dátami, či už lokálne alebo vzdialene. Mnohé emailové aplikácie dokážu spolupracovať s týmito nástrojmi a spoliehajú tak na ich možnosti. Synchronizačné aplikácie existujú vo forme samostatnej aplikácie (rSync) alebo vo forme prídavných rozšírení (napr. Thunderbird addons).

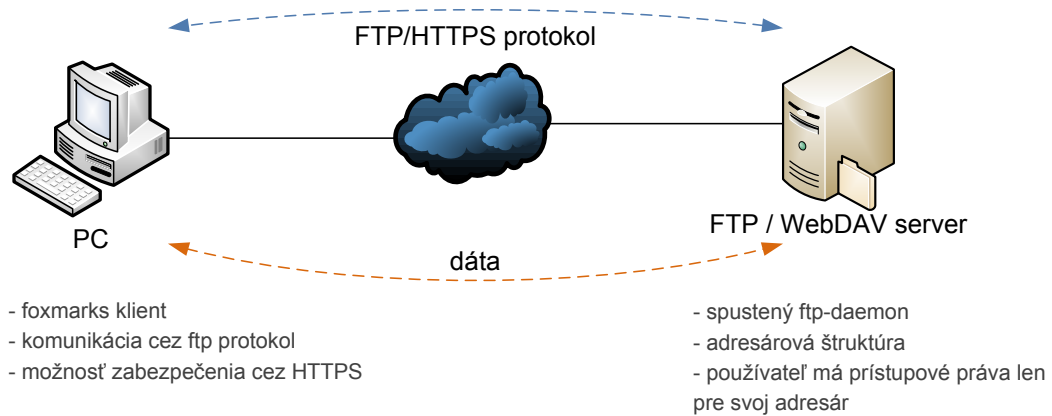
Základnú funkciu, ktorú podporuje väčšina emailových aplikácií je práca s adresárovými službami. Cez LDAP protokol možno pristupovať k adresárovým službám. Emailový klient plní funkciu rozhrania pre prístup k dátam. Na pozadí klient komunikuje so serverom posílaním vhodných príkazov. Používateľovi vráti zoznam kontaktov vyhovujúcich zadanému dotazu. Nevýhodou môžu byť pridelené prístupové práva tj. obmedzenia prístupu na čítanie/zápis. Riešenie je vhodné využiť v prípade, že sa jedná o centrálnu správu dát a tieto dáta sa nemenia často. Princíp činnosti je demonštrovaný na obr. 2.2.



Obrázek 2.2: Princíp LDAP adresára

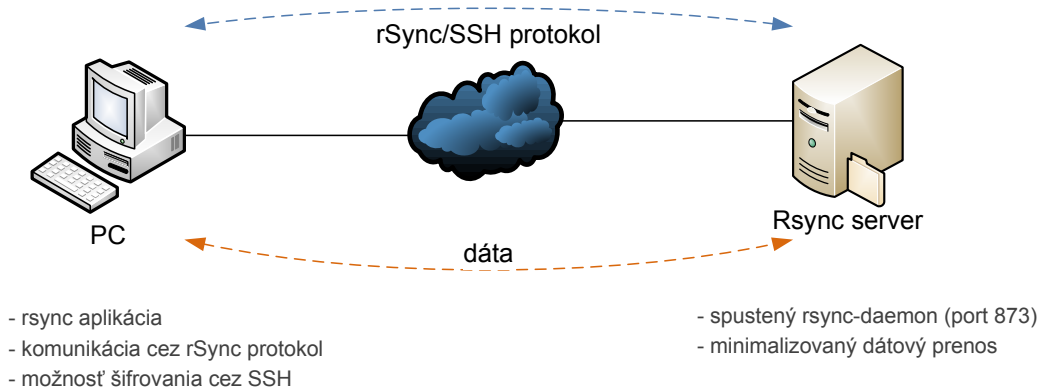
Zaujímavou službou z pohľadu synchronizácie je Foxmarks. Hoci nie je priamo určená na prácu s emailovými adresami, je vhodné ju spomenúť. Foxmarks slúži na synchronizáciu bookmarkov z Firefox internetového prehliadača. Je postavený na architektúre klient-server. Dáta sú prenášané pomocou HTTPS protokolu, čím je komunikácia zabezpečená. Na serveri beží WebDAV poprípade FTP server. Každý používateľ má pridelené prístupové práva k svojim vlastným dátam. Foxmarks podporuje aj zálohovanie na vlastný server, ktorého adresu zadáme v nastaveniach. Princípy fungovania by mohli byť uplatnené aj pri vlastnom návrhu synchronizácie adres. Výhodou tohto riešenia je univerzálnosť, rozšíriteľnosť a zabezpečenie prenášaných dát. Princíp fungovania tejto služby je zrejmy z obrázka 2.3.

Iné riešenie predstavuje synchronizácia pomocou rSync. RSync je aplikácia a protokol s vlastným algoritmom. Algoritmus je unikátny najmä v tom, že dokáže minimalizovať da-



Obrázek 2.3: Princíp Foxmarks synchronizácie

tový prenos. Synchronizovaný súbor sa rozdelí na viacero častí a prenášajú sa len zmeny. Prenášané súbory môžu byť navyše komprimované. Súbory sa prenášajú medzi dvoma zariadeniami v oboch smeroch. Jedno zo zariadení je tzv. "rsync-server" na ktorom beží rsync ako daemon. Na druhom zariadení je rsync spustený ako aplikácia. Komunikácia môže byť zabezpečená cez *ssh*. Na obrázku 2.4 je znázornený princíp synchronizácie.



Obrázek 2.4: Synchronizácia cez rsync

Kapitola 3

Vlastné riešenie

V tejto kapitole predstavím navrhnutý systém pre synchronizáciu emailových adries z vybraného poštového klienta a operačného systému.

3.1 Návrh systému

Vlastný návrh systému je založený na komunikácii klient-server. Klient je jednoduchá aplikácia na strane používateľa vo forme rozšírenia pre emailovú aplikáciu. Aplikácií na prácu s elektronickými adresami je veľké množstvo. Pre svoju prácu som zvolil emailového klienta Mozilla Thunderbird, ktorý je voľne dostupný. Výhodou toho programu je jeho otvorenosť. Program ponúka možnosť vytvorenia prídavných rozšírení a má k dispozícii plne zdokumentované API.

Druhú stranu komunikácie tvorí server. Klient podporuje komunikáciu so serverom prostredníctvom FTP a HTTPS protokolu. Na serveri je spustený FTP daemon, ktorý prijíma a ukladá dáta do príslušných adresárov. Každý používateľ má pridelený vlastný adresár, do ktorého ako jediného má plný prístup (zápis/čítanie). Správu používateľských účtov zabezpečuje MySQL. V databáze sú uložené prístupové práva, kvóty, prihlasovacie údaje, umiestnenie adresára a podobne. Prepojenie MySQL a FTP umožňuje *proFTPd* server.

Toto riešenie poskytuje dostatočnú univerzálnosť a možnosti rozširiteľnosti. Model systému je znázornený na obrázku 3.1.

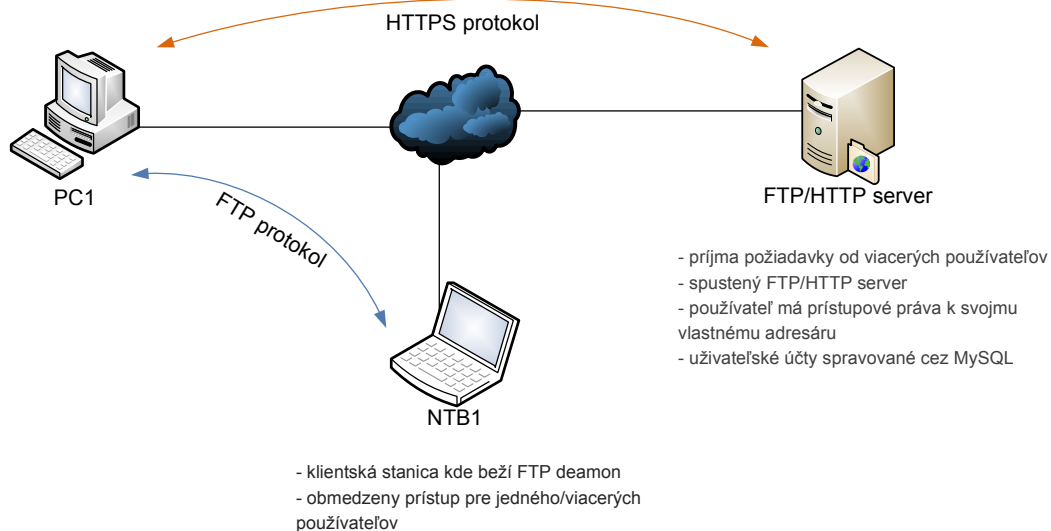
3.2 Implementácia

3.2.1 Klient

Klient je funkčným rozšírením emailovej aplikácie Mozilla Thunderbird. Aplikácia pozostáva z dvoch hlavných častí. Prvú časť tvorí grafické používateľské rozhranie (GUI) a druhú časť tvorí jadro aplikácie s funkciami zabezpečujúcimi chod aplikácie. GUI aplikácie je napísané v jazyku XUL a jadro aplikácie v jazyku JavaScript.

3.2.2 Štruktúra rozšírenia

Rozšírenie má predpísanú adresárovú štruktúru, ktorú je vhodné dodržiavať. Tým zabezpečíme jednoduchú inštaláciu a testovanie, pretože všetko rieši Add-on systém Thunderbirdu/Firefoxu. Výsledná adresárová štruktúra vrátane súborov je zabalená do ZIP archívu s príponou `.xpi`. Archív obsahuje:



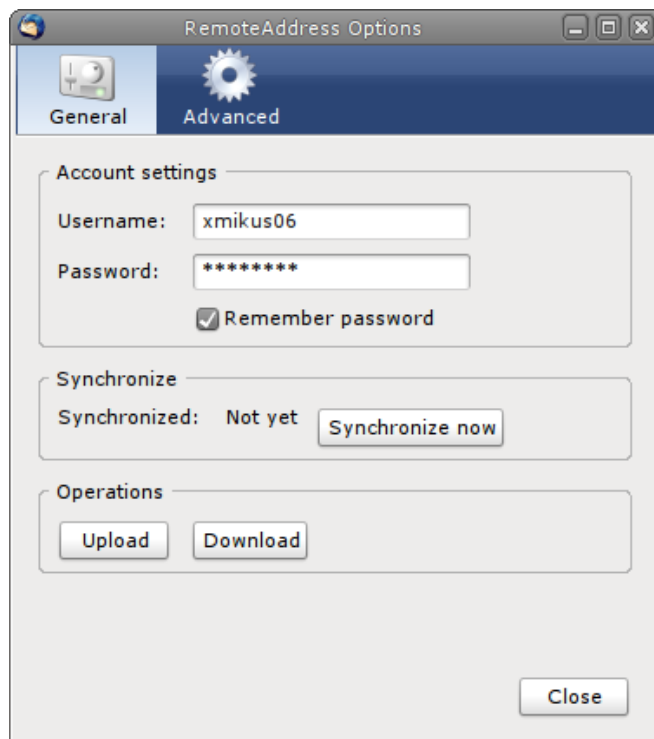
Obrázek 3.1: Návrh modelu systému

- `/install.rdf` - tento súbor obsahuje informácie (metadata) o rozšírení. Definuje XML menný priestor, špecifikuje verziu, autora, popis rozšírenia. Definuje aplikáciu, pre ktorú je rozšírenie určené (Thunderbird/Firefox) a minimálnu verziu tejto aplikácie.
- `/chrome.manifest` - súbor obsahuje cesty k jednotlivým elementom aplikácie. Napríklad k adresáru s nastaveniami, XUL rozhraniu, jazykovým lokalizáciam alebo vlastnej grafike.
- `/defaults/preferences/` - v adresári sa nachádza súbor `default.js` s predvolenými nastaveniami.
- `/chrome/locale/./` - adresár obsahuje jazykové mutácie. V súboroch sú uložené premenné a odpovedajúce texty GUI rozhrania v konkrétnom jazyku. V XUL rozhraniach sa stačí odkazovať na premenné. Add-on systém automaticky podľa nastavenia hlavného programu určí správny jazyk. Preklad rozšírenia je jednoduchý; stačí vytvoriť adresár odpovedajúci jazyku a preložiť súbor s dialógmi.
- `/skin/` - adresár obsahuje grafické prvky rozšírenia. Vzhľad XUL rozhrania je definovaný CSS súborom.
- `/chrome/content/` - jadro rozšírenia. Obsahuje súbory s XUL rozhraním a JavaScript súbory implementujúce hlavnú funkcionálnosť.

3.2.3 GUI

GUI rozšírenia tvoria `.xul` súbory. V súbore `remoteaddress-preferences.xul` som implementoval hlavné okno rozšírenia. Obsahuje štruktúru, z ktorej volám ďalšie dve pod-okná, tvoriace jednotlivé „záložky“. Toto riešenie je výhodné najmä v tom, že môže byť pohodlne rozšírené o ďalšie položky (záložky). Konkrétnym elementom sú priradené akcie (funkcie).

Funkcie sa nachádzajú v rovnomenných .js súboroch. Pri implementácii som využil previazanie s komponentou a službou *nsIPrefBranch*, ktorá pristupuje k hlavným nastaveniam aplikácie. Do XUL rozhrania sú tak načítané aktuálne nastavenia a pri korektnom ukončení sú nastavenia uložené. Týmto prístupom sú získané všetky potrebné parametre rozšírenia okrem mena a hesla. Tieto dve položky získavam cez službu *nsIPasswordManager*. Prihlasovacie meno a heslo je tak bezpečne uložené a centrálné spravované samotnou aplikáciou Thunderbird. Konečný vzhľad som dosiahol CSS súborom s definovaným štýlom. V okne zobrazujem aj čas poslednej úspešnej synchronizácie. Okno rozšírenia je na obrázku 3.2 a 3.3.



Obrázek 3.2: GUI rozšírenia (všeobecné nastavenia)

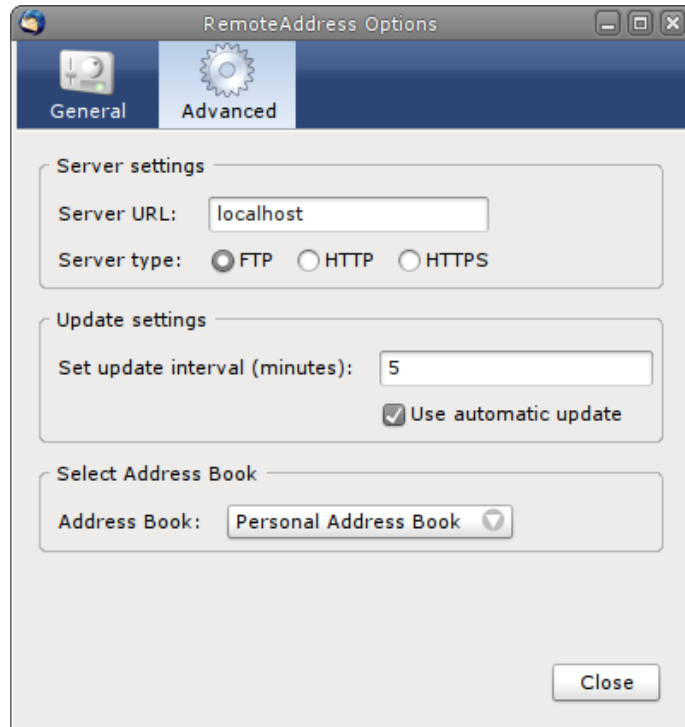
3.2.4 Ovládanie a inštalácia

Ovládanie je intuitívne, jednoduché a prehľadné. K nastaveniam programu sa pristupuje z Add-on správcu aplikácie Thunderbird, položka *options*. Hlavné okno rozšírenia tvoria dve záložky *General* a *Advanced*.

V záložke *General* sú umiestnené nastavenia mena a hesla k vzdialenému serveru a jednotlivé opeárcie *Synchronize*, *Upload*, *Download*. Meno a heslo je uložené bezpečne v centrálnej databázy hesiel programu Thunderbird. Operácia *Upload* a *Download* vykoná nútený Upload a Download bez ohľadu na to, či je na serveri novšia alebo staršia verzia adresára.

V záložke *Advanced* sú nastavenia týkajúce sa servera (adresa a cesta k súboru), interval automatickej synchronizácie a výber adresára, ktorý sa bude synchronizovať. Adresa musí končiť znakom */*.

Rozšírenie tvorí jediný XPI súbor. Inštaláciu rozšírenia zabezpečuje XPIInstall technológia, ktorá prebieha automaticky a plne v réžií Add-on systému aplikácie Thunderbird.



Obrázek 3.3: GUI rozšírenia (dodatočné nastavenia)

3.2.5 Synchronizácia

Synchronizácia môže byť automatická alebo manuálna. Manuálnu synchronizáciu spúšťa používateľ z hlavného okna rozšírenia. Manuálna synchronizácia zaručí, že vybraný adresár (súbor kontaktov) bude synchronizovaný s adresárom na serveri. Pri tomto type synchronizácie sa kontroluje čas poslednej modifikácie a prenáša sa len novší súbor.

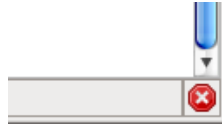
Pri automatickom type, sa spustený daemon stará o pravidelnú synchronizáciu v časových intervaloch zadaných používateľom v nastaveniach. Ak vyprší čas daemon zareaguje spustením funkcie synchronizácie. Synchronizácia samozrejme využíva porovnávanie času, a teda synchronizujú sa len novšie adresáre. Pri synchronizácii sa zavolá funkcia, ktorá má ako vstup parametre operácie. Funkcia následne vykoná potrebný upload/download a vráti chybový stav operácie.

Pre prístup k adresáru používam služby *nsIAbDirectory* (prístup k mozilla adresáru s kontaktami) a *nsILocalFile* (prístup k súborom na disku). Adresu potrebnú k pripojeniu na vzdialený server získam ako inštanciu objektu *URI*, ktorému ako parameter predám reťazec s adresou zostavenou z používateľského nastavenia. Zaujímavé na tomto objekte je, že aplikácia Thunderbird sama vie, na aký server sa má pripojiť a aké meno a heslo použiť pri autentifikácii. Adresa sa zostavuje v tvare `type://[user:pass@]host/cesta` (meno a heslo nie je povinné). Následne z adresy vytvorím komunikačný kanál službou *nsIIOService* a *nsIStringInputStream* do ktorého zapíšem vstupné dáta. Mozilla sa postará o samotný prenos dát. Na záver, v prípade funkcie upload, spracujem chybové hlásenia. V prípade funkcie download zmažem pôvodný adresár a vytvorím nový z prijatých dát.

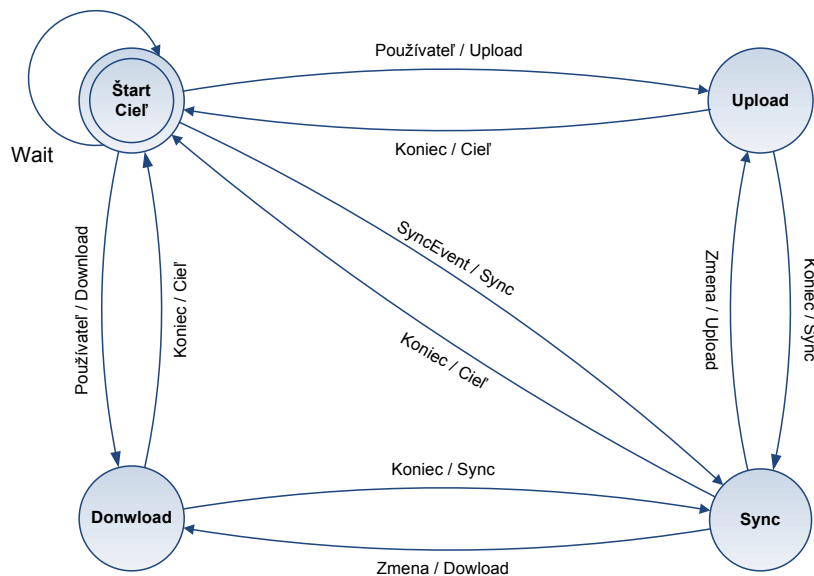
Automatický daemon využíva vstavanú metódu *addEventListener*, ktorá dovoľuje sledovanie udalostí. V prípade výskytu definovanej udalosti sa spustí jej obsluha. Metóda

umožňuje kontrolu nad výskytom udalostí a ich ošetrovanie. Zavádza sa automaticky po spustení programu a je prítomná počas jeho behu.

Indikáciou spusteného daemona a zároveň zobrazenie statusu vykonanej synchronizácie, je ikona v stavovom paneli hlavného programu (obr. 3.4).



Obrázek 3.4: Spustený daemon



Prechod z jedného stavu aplikácie do druhého (Vyvolaná akcia / Následujúci stav):

Používateľ – Akcia vyvolaná používateľom
Upload/Download – Akcia poslania/prijatia súboru
Koniec – Koniec operácie; vráti návratový kód
Štart/Cieľ – Počiatočný a koncový stav
Sync - Synchronizácia
SyncEvent - Akcia ktorá vyvolá synchronizáciu (časovač, používateľ)

Obrázek 3.5: Stavový automat synchronizácie

3.2.6 Server

Ako server možno použiť ľubovoľnú implementáciu FTP daemona. FTP daemon môže bežať na hostiteľskej stanici alebo na dostupnom serveri. Medzi najznámejšie implementácie patrí *proFTPd* (Linux), *FileZilla* (Windows) a iné. Pre potreby tejto práce som zvolil *proFTPd*, ktorý má oproti ostatným produktom niekoľko výhod. Server možno rozšíriť o moduly (napr. MySQL alebo TLS) a poskytuje pokročilú konfiguráciu. Využitím MySQL som získal pohodlnú správu viacerých používateľov. ProFTPd je dostupný pre operačný

systém linux, pre systém Windows existuje napríklad FileZilla Server. Na serveri musí používateľ vytvoriť konto a nastaviť k nemu prístupové heslo. Dôležité je tiež nastaviť prístupové práva k súborom tj. práva pre vlastníka súboru.

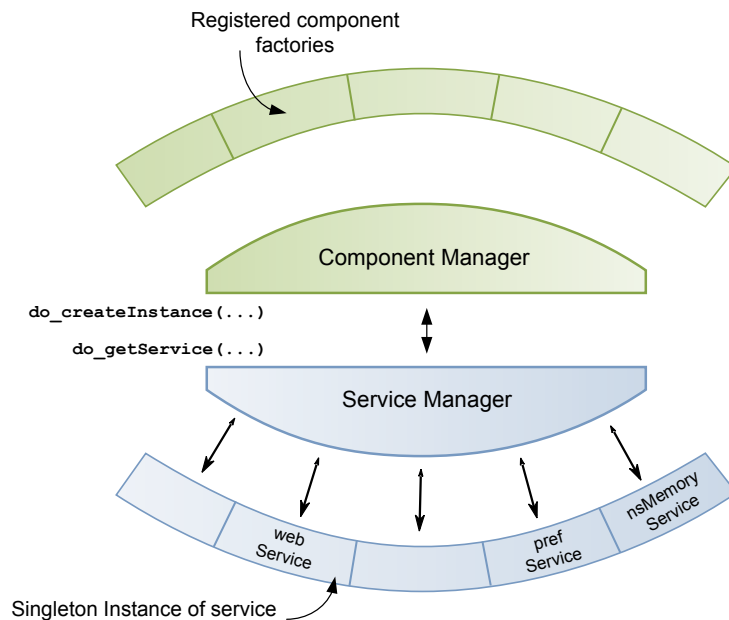
Ako HTTP server som použil inštaláciu *Apache*. Tento server beží ako na systéme Linux tak na systéme Windows. Pre bezchybné fungovanie je potrebné mať správne nakonfigurovaný SSL modul a nastavené prístupové práva. Pri testovaní sa mi osvedčila inštalácia XAMPP servera, ktorá obsahuje Apache vrátane SSL modulov a obsahuje aj FileZilla FTP server. XAMPP je dostupný aj ako zip archív, ktorý nevyžaduje inštaláciu.

3.3 Mozilla API

Mozilla API pozostáva z kolekcie služieb a komponentou, ktoré sú prístupné cez XPCOM rozhranie. Vrstva XPCOM je zložená s modelu komponent a infraštruktúry potrebnej k podpore dynamickej registrácie a manipulácie s XPCOM komponentami. Centrom XPCOM implementácie je *Service Manager* a *Component Manager*. Tieto dve služby spolu poskytujú centralizovaný bod, pre prístup ku všetkým verejným Mozilla rozhraniám.

Service Manager sprístupňuje všetky dostupné XPCOM služby - každá služba predstavuje objekt s určitou funkcionalitou. Component Manager vytvára nové inštancie registrovaných XPCOM komponent [11].

Veľká časť Gecko funkcionality je dostupná cez komponentu *nsWebBrowser*. Začlenené aplikácie (rozšírenia) môžu využiť túto komponentu k jednoduchému prístupu k Gecko funkciám. Každá inštancia WebBrowser reprezentuje tzv. „klientsku oblasť“ typického okna prehliadača. WebBrowser vystavuje rozhranie, ktoré dovoľuje aplikáciám kontrolovať aktivity a reagovať na zmeny v tejto oblasti.



Obrázek 3.6: Mozilla API

3.3.1 XPCOM

XPCOM je objektový model nezávislý na platforme, podobný Microsoft COM. Má podporu viacerých programovacích jazykov a umožňuje implementáciu komponent v Javascripte, Jave, Python a C++. XPCOM poskytuje v základe množinu komponentou a tried, napríklad správu súborov a pamäti, vlákna, datové štruktúry, . . . Väčšina komponentou však nieje súčasťou jadra a je dodaná inými časťami, aplikáciami alebo dokonca rozšíreniami.

3.3.2 XML User Interface Language

Používateľské rozhranie programu Mozilla Thunderbird (Firefox) je napísané v XUL a Javascripte. XUL je XML gramatika, ktorá poskytuje používateľské widgety ako napríklad tlačítka, panel nástrojov, rozbalovacie ponuky a ďalšie. Jednotlivé akcie vyvolané rozhraním, vykonáva JavaScript. Výhodou XUL je prenositeľnosť a platformová nezávislosť.

3.3.3 XULRunner

XULRunner je runtime balík, ktorý je použitý na zavedenie XUL+XPCOM plnohodnotných aplikácií podobných Firefoxu a Thunderbirdu. Dáva k dispozícii mechanizmus pre inštaláciu, aktualizáciu a odinštaláciu týchto aplikácií. Poskytuje tiež riešenie *libxul*, dovoľujúce začlenenie Mozilla technológie do iných projektov a produktov.

3.3.4 XPInstall

Cross-Platform Install (XPI) je technológia používaná v Mozilla Thunderbird/Firefox alebo iných XUL-založených aplikáciách, na tvorbu inštalačných mechanizmov pre rozšírenia. XPI inštalačný modul je ZIP archív, ktorý obsahuje inštalačný skript (.js) alebo manifest (.rdf) v koreňovom adresári tohto archívu. Inštalácia rozšírenie tak prebieha automaticky v réžií Add-on systému programu.

3.3.5 Vývojové nástroje

Vývojových nástrojov existuje celá rada. Medzi doporučené nástroje, ktorými možno vytvárať rozšírenia, patria programy ako *DOM Inspector*, *Venkman* (JavaScript debugger), *Spket IDE* alebo *NetBeans IDE* (s pluginom Foxbeans). Pri vývoji rozšírenia som používal textový editor so zvýrazňovaním syntaxe PSPad (Windows) a Kate (Linux). Je užitočné mať správne nastavené prostredie programu Thunderbird/Firefox. Doporučené nastavenie prostredia:

- `javascript.options.showInConsole = true` - loguje chyby počas behu programu a výstup je presmerovaný na chybovú konzolu (*Error Console*).
- `nglayout.debug.disable_xul_cache = true` - vypína vyrovnávaciu pamäť (cache) pre XUL rozhranie. Možno tak editovať súbory bez nutnosti reštartovať hlavný program.
- `browser.dom.window.dump.enabled = true` - zapína možnosť použiť funkciu `dump()`. Táto funkcia slúži k výpisu hlásení na konzolu.
- `javascript.options.strict = true` - nastavenie povoľuje výpis chýb na chybovú konzolu. Produkuje veľké množstvo hlásení a chýb.

- `extensions.logging.enabled = true` - posiela detailný výpis problémov vzniknutých v priebehu inštalácie a aktualizácie na chybovú konzolu.

Rozšírenie nemusí byť počas samotného vývoja inštalované klasickou metódou cez Add-on manažér. Rozšírenie (súbor `.xpi`) stačí nakopírovať priamo do adresára programu (`<appdir>/extensions/<extensionID>`), odkiaľ sa pri jeho spustení samo rozbali a nainštaluje. Tým sa urýchli zdlhavý proces inštalácie cez Add-on manažér.

3.4 Bezpečnosť

Nastavenia Všetky nastavenia (premenné) rozšírenia sú uchovávané a spravované centrálné programom Thunderbird. Tieto nastavenia sú voľne dostupné napríklad cez rozhranie `about:config`. Cez toto rozhranie možno ľubovoľne meniť hodnoty premenných.

Z tohto dôvodu som sa rozhodol hlavné meno a heslo k vzdialenému účtu uložiť mimo túto databázu. Mozilla poskytuje okrem hlavnej databázy, špeciálnu oblasť pre ukladanie a správu hesiel. Táto oblasť má možnosť, podľa nastavenia používateľa, nastavenia hlavného hesla. Pri každom pokuse o prístup do tejto oblasti sa program zareaguje výzvou na zadanie hlavného hesla (pokiaľ je nastavené). Pre prístup k menu a heslu používať službu *nsIPasswordManager*.

Samotné rozšírenie ponúka možnosť neuloženia hesla a spolieha tak na ručne vyplnené hodnoty. V tomto prípade však automatická synchronizácia (daemon) nie je k dispozícii.

Komunikácia Ako bolo naznačené v úvode tejto kapitoly, rozšírenie podporuje komunikáciu cez HTTPS protokol. Protokol HTTPS je šifrovanou verziou protokolu HTTP. Klient (thunderbird rozšírenie) sa autentizuje na vzdialenom servery pomocou mena a hesla zadaného v nastaveniach programu. Komunikáciu zabezpečuje služba *nsIHttpChannel*.

3.5 Testovanie

Aplikácia podporuje výpis chybových hlásení, ktoré môžu vzniknúť počas jej používania. Testuje chyby v používateľských nastaveniach a rovnako testuje chyby vzniknuté počas nadväzovania spojenia medzi serverom a klientom. V prípade výskytu chyby aplikácia zobrazí chybové hlásenie a ukončí svoj beh, tzn. nedojde k zničeniu dát na strane klienta.

Aplikácia v podobe rozšírenia bola testovaná v programe Thunderbird vo verzií 2.0.0.14 v prostredí Windows a v prostredí Linux (Ubuntu). Ako server bol pri testovaní použitý XAMPP server (FTP, HTTP, HTTPS). Synchronizácia ako aj manuálny upload/download prostredníctvom podporovaných protokolov (FTP, HTTP, HTTPS) prebehol úspešne. Testované dáta boli porovnané s dátami uloženými na serveri a v prípade rozdielných verzií bol zahájený samotný prenos. Pri rovnakej verzií dát nedojde k ich prepísaniu. V prípade nekonzistentnosti dát na serveri (neúplný upload, poškodené dáta), aplikácia vypíše chybu a skončí.

3.6 Možnosti rozšíriteľnosti

Navrhnutý systém poskytuje niekoľko možností rozšíriteľnosti, ktoré by mohli byť dodatočne implementované.

Jednou z možností je vytvorenie webovej aplikácie, ktorá by spravovala používateľské účty. To je možné dosiahnuť použitím *proFTPD* v spojení s MySQL serverom. Webová aplikácia by registrovanému používateľovi ponúkala správu svojich emailových adries a kontaktov priamo prostredníctvom webového rozhrania. Toto riešenie poskytuje mobilitu, dostupnosť a správu kontaktov odkiaľkoľvek.

Ďalšia možnosť rozšíriteľnosti spočíva vo využití algoritmu, podobného *rSync* algoritmu. Prínos by spočíval v prenášaní zmien nad dátami, čím by sa minimalizovala prevádzka.

Rozšírenie by bolo možné doplniť o podporu ďalších protokolov, napríklad *rSync* alebo *syncML*. Tiež by bolo možné rozšírenie upraviť aby podporovalo niektoré dostupné služby napríklad *Plaxo* alebo *Foxmarks*.

V tabuľke 3.1 je uvedený prehľad funkcií a vlastností podporovaných vytvoreným rozšírením a možnosti rozšírenia projektu.

Vlastnosti	Implementácia	Možnosť rozšírenia
Podporovaný protokol	FTP, HTTP, HTTPS	LDAP
Spôsob synchronizácie	Prenášaný celý obsah súboru	Využitie diferenciálneho prenosu
Formát prenášaného súboru	Mozilla Address Book	Univerzálny súbor (napr. XML)
Bezpečnosť	Spolieha na prenosový protokol	Šifrovanie súboru počas prenosu
Automatická synchronizácia	V časových intervaloch	
Kontrola konzistentnosti dát	MD5 kontrolný súčet	

Tabuľka 3.1: Prehľad vlastností a funkcií podporovaných rozšírením

Kapitola 4

Záver

Cieľom tejto bakalárskej práce bolo zaoberať sa spôsobmi ukladania emailových adries poštovými programami, možnosťami ukladania adries na centrálnom serveri ako aj dostupnými možnosťami synchronizácie súborov.

Na začiatku som sa venoval teoretickému pohľadu na prácu s emailovými adresami a synchronizácií súborov. Na vybraných klientoch som uviedol spôsoby ukladania emailových adries. Boli predstavené dostupné možnosti synchronizácie. Pri jednotlivých riešeniach som poukázal na ich výhody a nevýhody. Tieto vlastnosti ovplyvnili návrh vlastného modelu.

V druhej kapitole som navrhol a popísal vlastný model systému určený k synchronizácií emailových adries. Ako cieľového klienta pre rozšírenie som zvolil program Mozilla Thunderbird a implementačný jazyk JavaScript. Hlavným argumentom bola jeho otvorenosť a prenositeľnosť. Pri návrhu som vychádzal z podobných dostupných riešení. Rozšírenie komunikuje s FTP alebo HTTP serverom. Uviedol som popis jednotlivých častí systému a popísal som implementáciu. Stručne som predstavil Mozilla vývojové rozhranie.

Navrhnuté riešenie plní svoju úlohu v oblasti synchronizácie. Systém poskytuje dostatočnú úroveň zabezpečenia. Systém ako aj samotná aplikácia je univerzálny a jeho funkcionality môže byť dodatočne rozšírená. Klient aj server sú platformovo nezávislé a môžu byť nasadené nad operačným systémom Linux alebo Microsoft Windows.

Literatura

- [1] Carter, G.: *LDAP System Administration*. O'Reilly, 2003, iSBN 1-56592-491-6.
- [2] Dawson, F.: vCard MIME Directory Profile. RFC 2426, September 1998.
URL <http://tools.ietf.org/html/rfc2426>
- [3] Good, G.: The LDAP Data Interchange Format (LDIF) - Technical Specification. RFC 2849, Jún 2000.
URL <http://tools.ietf.org/html/rfc2849>
- [4] Mahmoud, Q. H.: Getting Started with Data Synchronization Using SyncML. September 2004.
URL <http://developers.sun.com/mobility/midp/articles/syncml/>
- [5] Shafranovich, Y.: Common Format and MIME Type for Comma-Separated Values (CSV) Files. RFC 4180, Október 2005.
URL <http://tools.ietf.org/html/rfc4180>
- [6] Sitera, J.: Technická zpráva TEN-155 CZ číslo 4/2000. Technická zpráva, September 2000.
URL <http://www.cesnet.cz/doc/techzpravy/2000-4/>
- [7] Tridgell, A.; Mackerras, P.: The rsync algorithm. Technická zpráva, Australian National University, Australia, November 1998.
URL http://rsync.samba.org/tech_report/
- [8] Versit-Consortium: The Electronic Business Card. Technická zpráva, September 1996.
URL <http://www.imc.org/pdi/vcard-21.txt>
- [9] File synchronization. In *Wikipedia - The Free Encyclopedia*, nadace WIKIMEDIA, September 2007.
URL http://en.wikipedia.org/wiki/File_synchronization
- [10] WWW-stránky: Foxmarks synchronizer. <http://www.foxmarks.com>.
- [11] WWW-stránky: Mozilla Developer Central. <http://www.mozilla.org/developer/>.