

**Ján Garaj**

**OPTIMALIZÁCIA VYUŽITIA VÝPOČTOVÝCH  
PROSTRIEDKOV PRI E-MAILOVEJ KOMUNIKÁCII**

**Bakalársky projekt**



Vedúci bakalárskeho projektu:  
Ing. Ján Máté  
máj, 2007

Čestne prehlasujem, že som bakalársky projekt vypracoval samostatne  
s použitím uvedenej literatúry a zdrojov.

Bratislava, 10.mája 2007

Ján Garaj

~~zadanie — kopia resp./original~~

## **ANOTÁCIA**

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Študijný program: Informatika

Autor: Ján Garaj

Bakalársky projekt: Optimalizácia využitia výpočtových prostriedkov pri e-mailovej komunikácii

Vedenie bakalárskeho projektu: Ing. Ján Máté  
máj, 2007

Úlohou projektu je optimalizovať využitie výpočtových prostriedkov pri emailovej komunikácii. Východiskom práce je analýza súčasnej implementácie typického emailového servera so zameraním na využitie diskovej kapacity. Cieľom je vytvorenie a implementácia mailového softvéru, ktorý by eliminoval nevýhody vlastností dnešných mailových systémov.

## **ANNOTATION**

Slovak University of Technology Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree Course: Informatics

Author: Ján Garaj

Bachelor Theses: Optimization of the Usage of Computer Systems In E-mail  
Communication

Supervisor: Ing. Ján Máté

2007, May

The aim of the theses is to optimize the usage of computer systems in e-mail communication. The starting point of it is the analysis of the contemporary implementation of the common e-mail server with the stress laid on the usage of its disc capacity. The aim is to design and implement a e-mail software that would eliminate the negatives of contemporary e-mail system.

# Obsah

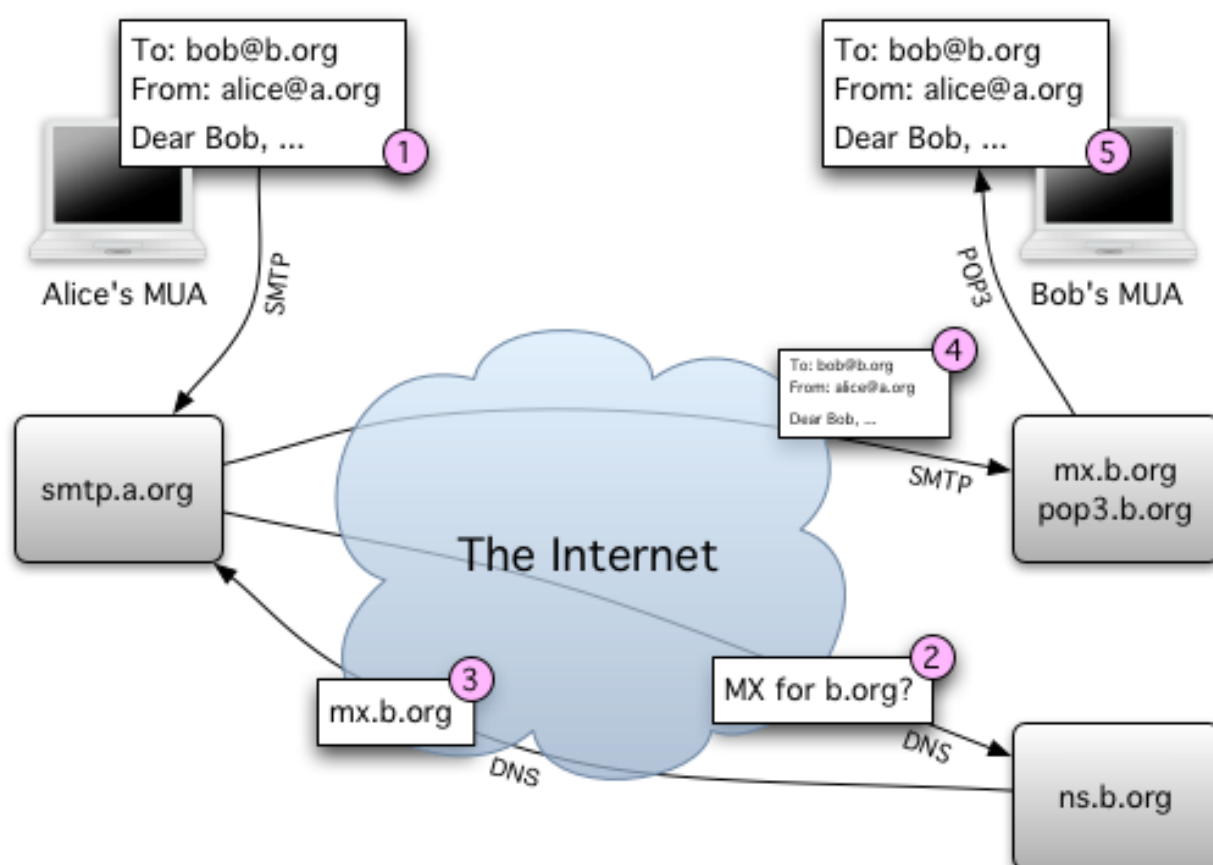
<b>1 Úvod.....</b>	<b>7</b>
1.1 Elektronická pošta.....	7
1.2 Ako pracuje email.....	7
1.3 Súčasný stav emailovej komunikácie.....	9
1.4 Porovnanie emailových serverov.....	10
<b>2 Analýza problémovej oblasti.....</b>	<b>13</b>
2.1 Emailový server.....	13
2.2 Emailový server Postfix.....	14
2.3 Spôsoby ukladania emailov.....	15
2.4 Efektívnosť využívania diskovej kapacity serverom Postfix.....	16
2.5 Špecifikácia požiadaviek.....	17
<b>3 Návrh riešenia.....</b>	<b>17</b>
3.1 Možné spôsoby riešenia.....	17
3.2 Linky - hardlink a symlink, iné riešenia .....	18
3.3 Prístupové práva.....	20
3.4 Integrácia s emailovým serverom Postfix.....	23
3.5 Vnútročné členenie optimalizačného modulu.....	24
<b>4 Technická dokumentácia.....</b>	<b>26</b>
4.1 Implementácia optimalizačného modulu.....	26
4.2 Diagram činností.....	27
4.3 Popis činnosti.....	28
4.4 Inštalácia a požiadavky.....	31
4.5 Overenie riešenia.....	34
<b>5 Zhodnotenie.....</b>	<b>34</b>
<b>6 Záver.....</b>	<b>34</b>
<b>7 Slovník pojmov problémovej oblasti.....</b>	<b>35</b>
<b>8 Zoznam použitej literatúry.....</b>	<b>37</b>
<b>9 Príloha A.....</b>	<b>39</b>

## 1 Úvod

### 1.1 Elektronická pošta

Elektronická pošta (skrátенý názov e-mail alebo email z angl. electronic mail) je metóda na tvorbu, posielanie, ukladanie a prijímanie správ cez elektronický komunikačný systém. Výraz email označuje dve veci. V prvom rade Internetový systém doručovania založený na protokole SMTP (Simple Mail Transfer Protocol) [4] a taktiež intranetový systém umožňujúci používateľom jednej organizácie posielanie správ ostatným členom danej organizácie. Často sa na komunikáciu pomocou správ v rámci organizácie používa Internetový protokol namiesto intranetového systému a tak „email“ časom nadobudol význam obidvoch pojmov.

### 1.2 Ako pracuje email



Obrázok 1: Schéma udalostí pri posielaní a doručovaní emailu

Typický sled udalostí, emailovej komunikácie (príklad Alica píše email Bobovi) podľa obrázku 1:

1. Alica na vytvorenie emailu používa emailového klienta - Mail User Agent (ďalej ako MUA). Aby každý článok komunikácie vedel kde má smerovať je nutné už pri tvorbe emailu zadať položku „To“, ktorá určuje adresáta emailu. Taktiež sa zadáva aj položka „From“, aby adresát vedel určiť odosielateľa. Samozrejmosťou súčasťou emailu je aj správa, ktorú používateľ posielajú príjemcovi emailu. Po vyplnení položiek emailu a odoslání emailu používateľom sa emailový klient spojí protokolom SMTP so SMTP serverom, ktorý má uložený vo svojej konfigurácii a odošle email serveru.
2. SMTP server sa u prijatého emailu rozhoduje podľa hlavičky emailu. V hlavičke emailu nájde položku „To“ a podľa jej hodnoty sa rozhodne o ďalšom spracovaní emailu. V prípade, že SMTP server nie je emailovým serverom adresáta, server sa snaží doručiť email ďalej až k serveru adresáta. Aby mohol server túto akciu vykonať, potrebuje vedieť kto je emailovým serverom adresáta. V tejto chvíli sa dostáva ku slovu DNS protokol. SMTP server vygeneruje požiadavku na Name Server adresátovej domény so žiadosťou o MX (emailový) záznam pre doménu adresáta.
3. Name Server je server, ktorý má informácie o doménových a subdoménových menách a taktiež aj o emailovom serveri danej domény (domén), ktorú spravuje. V prípade požiadavky poskytuje informácie o menách a IP adresách požadovaných doménových mien, čo sa stane aj v tomto prípade.
4. Po získaní názvu emailového servera adresáta SMTP server pošle SMTP protokolom email na daný server. Typicky sa email doručuje prostredím Internetu. Cestou Internetom môže dôjsť k spracovaniu emailu MTA servermi, ktoré email posielajú k emailovému serveru adresáta, čiže cieľovému serveru. V prípade, že po doručení emailu na cieľový server adresát emailu existuje, server uloží email do odkladacieho priestoru adresáta.
5. Adresát (Bob) používa emailového klienta, ktorý sa v pravidelných intervaloch pripája protokolom POP3 [5] na emailový server. V prípade uloženého mailu na serveri, stiahne daný email na lokálny počítač a sprístupní ho adresátovi.



### 1.3 Súčasný stav emailovej komunikácie

Analýza súčasnej situácie vychádza z prieskumu Regentskej univerzity, ktorá sa zaoberala Internetom vo svojej práci *How Much Information? 2003* [13]. Uvedená práca cituje spoločnosť International Data Corporation, zaoberajúcu sa marketingovými prieskumami. Táto spoločnosť odhadla objem priemernej dennej celkovej emailovej komunikácie na rok 2006 na 60 miliónov emailov denne. V prieskume sa uvádza aj spoločnosť Forrester Research, ktorá odhadla priemernú veľkosť emailu na 59KB. Vychádzajúc z uvedených čísel vychádza denne viac ako 300TB emailovej celosvetovej komunikácie. Uvedené čísla poukazujú na vysoký stupeň využívania emailov zo strany používateľov Internetu. Na uvedené počínanie používateľov reagovali aj poskytovatelia prístupu na Internet, ktorý pre svojich klientov zriaďovali a ešte zriaďujú emailové servery. Celosvetový počet mailových serverov je dnes vyše 1,5 milióna. Tabuľka 1 [15] ukazuje v akom pomere sú aktuálne (november 2006) zastúpené jednotlivé druhy emailových serverov.

Typ emailového servera	Počet serverov	Percentuálny podiel
Sendmail	297 343	32.90%
Microsoft Exchange	187 955	20.79%
Exim	158 381	17.52%
Postfix	115 219	12.75%
IMail	46 791	5.18%
MDaemon	21 071	2.33%
MailEnable	18 169	2.01%
ostatné	58 929	6.50%

Tabuľka 1: Rozdelenie používaných mailových serverov vo svete podľa typu

## 1.4 Porovnanie emailových serverov

V nasledujúcom porovnaní sa budem zaoberať iba prvými štyrmi emailovými servermi uvedených v tabuľke 1, nakoľko majú spoločne takmer 85% majoritný podiel na celkovom trhu emailových serverov.

Každý z porovnávaných serverov má svoje výhody, vie spracovať veľké množstvo emailov, spolupracovať s rôznymi databázami, má svoju dokumentáciu a možno aj vlastnosti, ktoré konkurenčný softvér nemá.

Celkovo bude porovnanie zamerané na nasledujúce kritéria:

- ♦ jednoduchosť administrácie
- ♦ bezpečnosť
- ♦ výkon
- ♦ ostatné

Všetky porovnania sa odkazujú na výsledky uvedené na obrázku 2 [17] a v tabuľke 2 [18]. Obrázok 2 ukazuje výsledky vybraných serverov v základnej konfigurácii (t.j. bez integrovaného antivírusového systému) a taktiež v konfigurácii s antivírusovým systémom na rôznych súborových systémoch. Tabuľka 2 obsahuje výsledky konkrétneho aplikačného testu.

*Sendmail* [2] [11]

- ♦ jednoduchosť administrácie – výhodou je jeden konfiguračný súbor, ktorý je však na druhej strane veľmi zložitý, nakoľko sa používa pomerne zložitý makrojazyk M4.
- ♦ bezpečnosť – od počiatku existencie Sendmailu, roku 1981, dodnes bolo v Sendmaily objavených niekoľko desiatok závažných chýb, ktoré sú archivované v rôznych databázach chýb po celom svete ako sú National Vulnerability Database alebo Open Source Vulnerability Database.
- ♦ výkon – v základnej konfigurácii (bez antivírusu) má spomedzi testovaných OpenSource serverov relatívne najnižšiu priepustnosť emailov
- ♦ ostatné – z historického hľadiska bol Sendmail prednastaveným emailovým serverom vo viacerých Linuxových distribúciách a aj vďaka tomuto faktoru je ešte stále veľmi nasadzovaný. K výhodám Sendmailu patrí efektívna implementácia tzv. „*militer*“ filtrov, voľne prekladaných ako „mailových filtrov“, ktoré sa využívajú ako rozhranie pre softvér tretích strán a taktiež port na Windows. Nevýhodou je monolitickosť servera.

### *Microsoft Exchange* [14]

- ♦ jednoduchosť administrácie – používateľsky prívetivé grafické rozhranie, od verzie 2007 dostupné aj skriptovacie rozhranie Windows PowerShell
- ♦ bezpečnosť – databázy chýb softvéru archivujú niekoľko desiatok bezpečnostných chýb. Microsoft však pravidelne vydáva updaty na tento svoj produkt.
- ♦ výkon – podľa dostupných testov [19] je jeho emailová priepustnosť porovnateľná s ostatnými emailovými servermi
- ♦ ostatné – Exchange je možné nasadiť ako groupwarové riešenie pre firmy. V spolupráci s ďalšími produktmi Microsoftu vie „tlačiť“ emaily aj na bezdrôtové prístroje siete GSM, tzv. „BlackBerry“. Nevýhodou je uzavretý zdrojový kód a proprietárna licencia k produktu. Formát ukladania emailov je iba do databázy, avšak má implementovanú funkciu „*single instance storage*“ [16], ktorá ukladá do databázy jednu správu určenú viacerým adresátom iba raz.

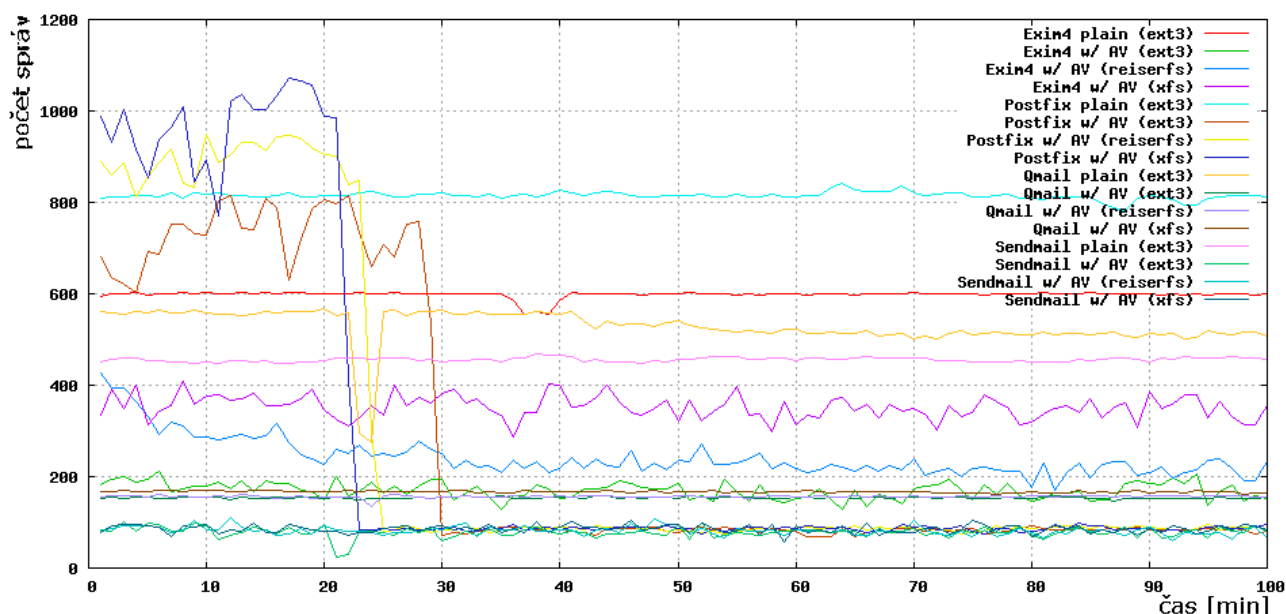
### *Exim* [12]

- ♦ jednoduchosť administrácie – jednoduchá, obsahuje iba jeden konfiguračný súbor
- ♦ bezpečnosť – necelá desiatka chýb hodnotených ako vysoko nebezpečné svedčia o celkovo dobrej bezpečnosti
- ♦ výkon – z testovaných serverov sa umiestnil zhruba v strede, vyniká svojou časovou stabilitou pri spolupráci s antivírusovým systémom.
- ♦ ostatné – Exim má otvorený zdrojový kód je možné ho spustiť aj pod operačným systémom Windows použitím Cygwin emulátoru. Pri správnom nastavení je možné volať perl interpreter priamo z konfiguračného súboru. Exim má aj vlastný filtrovací jazyk. Nevýhoda je monolitickosť tohto emailového servera.

## 1.4 Porovnanie emailových serverov

### Postfix [1] [10]

- ♦ jednoduchosť administrácie – priemerná, obsahuje niekoľko konfiguračných súborov, z ktorých sa však edituje zvyčajne iba jeden
- ♦ bezpečnosť – počet bezpečnostných chýb od vzniku tohto servera sa pohybuje mierne nad 10, takže je všeobecne považovaný za bezpečný
- ♦ výkon – vo väčšine kategórií mu patri prvenstvo a taktiež aj v celkovom hodnotení je najvýkonnejší, t.j. má najväčšiu priepustnosť emailov
- ♦ ostatné – používatelia oceňujú najmä jednoduchosť a bezpečnosť, ktorú autor matematik Wietse Venema, dosiahol modulárnosťou servera na rozdiel od zaužívaných monolitických riešení. Nevýhodou je neexistencia verzie na OS Windows.



Obrázok 2: Priebeh priepustnosti vybraných OpenSource emailových serverov na rôznych súborových systémoch a s rôznymi konfiguráciami

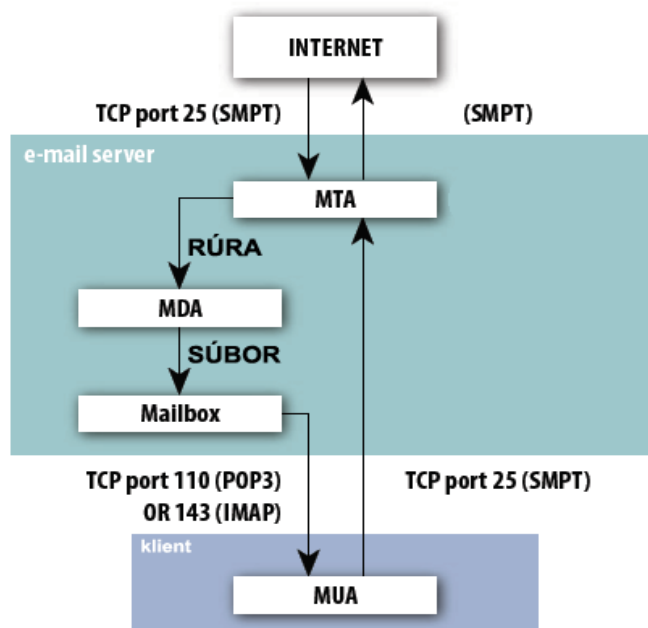
server	trvanie spojenia [s]	trvanie doručenia [s]	sync zápis	async zápis	priepustnosť [mail/s]	poznámka
1000 emailov pre 1 adresáta s 20 vláknami, 50 emailov na spojenie						
exim	56	262	6,621	6,785	3.8	manual queue run unsafe
postfix	49	62	1,898	4,110	16.1	-
sendmail	136	136	8,172	7,993	7.3	no softupdates FreeBSD 4.4-RC

Tabuľka 2: Testovanie vybraných OpenSource emailových serverov

## 2 Analýza problémovej oblasti

### 2.1 Emailový server

Vzhľadom k výkonu a vzhľadom k všetkým spomenutým vlastnostiam emailového servera Postfix, vrátane modularity, otvoreného zdrojového kódu a jednoduchej konfigurácii sa v ďalších častiach budem podrobne zaoberať emailovým serverom Postfix. Vnútroštruktúru Postfixu, ktorá platí všeobecne pre väčšinu emailových serverov na operačnom systéme UNIX, znázorňuje obrázok 3 [3]. Vždy je prítomný Mail Transfer Agent (ďalej označovaný skratkou MTA), ktorý zabezpečuje SMTP komunikáciu smerom do a z Internetu a k emailovým klientom. MTA počúva na TCP porte číslo 25, ktorý je určený na SMTP komunikáciu.

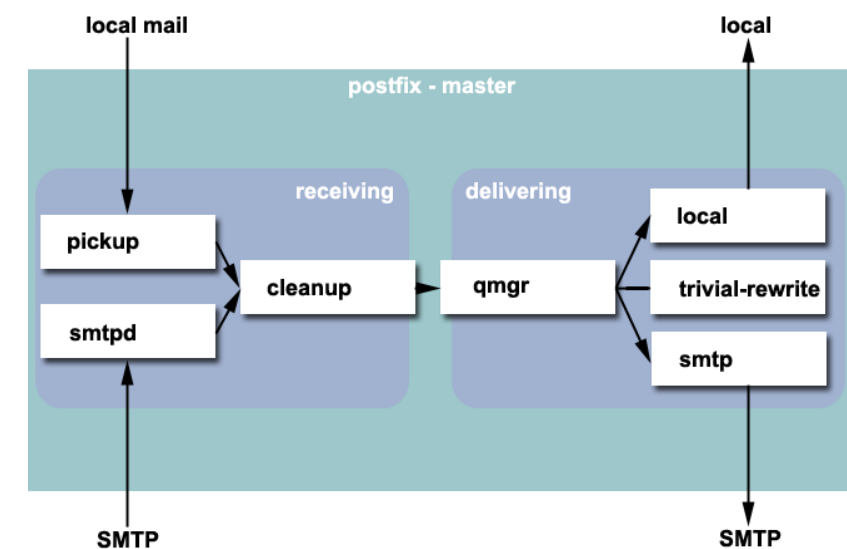


Obrázok 3: Schéma emailového systému na OS UNIX

Ďalšou súčasťou Postfixu je Mail Delivery Agent (ďalej označovaný skratkou MDA), ktorého úlohou je doručovanie emailov do lokálnych emailových adresárov, prípadne do mailboxov používateľov. Z MTA zvyčajne získava emaily cez dátovod (rúru) a jeho výstupom je už konkrétny súbor v emailovom adresári. Postfix má v konfiguračnom súbore direktívu, ktorá umožňuje použitie externého MDA, čo bude pravdepodobne užitočná vlastnosť pre moju ďalšiu prácu.

## 2.2 Emailový server Postfix

Postfix je implementovaný ako jeden hlavný server (proces "*master*"), ktorý spúšťa obslužné démoni vykonávajúce špecifické operácie podľa potreby. Jednotlivé služby sa spúšťajú ako samostatné procesy a zabezpečujú napr. posielanie a prijímanie mailov z Internetu, doručovanie lokálnych mailov atď. Počet procesov, ktoré daný typ požiadavky obsluhujú, je určený konfiguráciou.



Obrázok 4: Členenie a spolupráca modulov v serveri Postfix

Stručný popis funkcie démonov servera Postfix podľa obrázku 4:

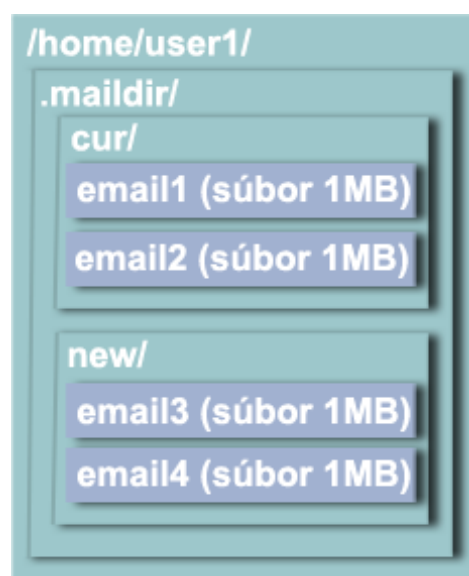
- ♦ *smtpd*: počúva na porte a prijíma SMTP požiadavky. Všetky prijaté správy sú presmerované na démona "*cleanup*"
- ♦ *pickup*: čaká na lokálne napísané emaily a presmeruje ich obsah na démona "*cleanup*"
- ♦ *cleanup*: spracúva prijatý mail (pridáva chýbajúce hlavičky a pod.), vkladá ho do fronty prijatých emailov a informuje démona "*qmgr*" o jeho príchode
- ♦ *qmgr*: čaká na prijaté emaily a zabezpečuje ich doručenie. Spôsob doručenia určí démon "*trivial-rewrite*"
- ♦ *trivial-rewrite*: prepisuje adresu do štandardizovanej formy. Démon pripája meno domény k lokálnym emailom bez jej uvedenia a pod. Okrem toho určuje, čo sa stane s emailom, ako a kam sa bude doručovať na základe adresy
- ♦ *local*: doručuje mail do lokálnych schránok na serveri
- ♦ *smtp*: smtp klient Postfixu. Doručuje maily z mailovej fronty, ktoré sú určené pre iné mailové servery.

## 2.3 Spôsoby ukladania emailov

Ako už bolo spomenuté všetky emaily sa používateľom ukladajú na súborový systém, buď do mailboxu alebo do maildiru. Rozdielna štruktúra mailboxu a maildiru je načrtnutá na obrázkoch 5 a 6. Ďalšia možnosť je využitie databázy ako úložisko mailov. Výhoda tohto riešenia je v jednoduchšej manažovateľnosti a flexibilitnosti, avšak celý emailový systém je plne závislý od vlastností a spoľahlivosti použitej databázy.



Obrázok 5: Štruktúra mailboxu



Obrázok 6: Štruktúra maildiru

V mailboxe sa emaily ukladajú sekvenčne v jednom súbore. Nevýhoda tohto spôsobu archivácie emailov sa objaví pri veľkom počte uložených emailov, keď klientský program má vyhľadať jeden konkrétny email. Čas trvania operácie bude priamo úmerný veľkosti celého mailboxu. Túto spomenutú nevýhodu eliminuje maildir tým, že pri ukladaní využíva hierarchické vlastnosti súborového systému. Konkrétne pri maildir-e sa jednotlivé emaily ukladajú do samostatných súborov, ktoré sa členia do adresárovej štruktúry. Výhodou tohto riešenia je, že pri vyhľadávaní a ostatných operáciách sa neprehľadáva obsah jedného súboru, ale prehľadáva sa adresárová štruktúra. K výhodám maildiov a taktiež mailboxov je možné ešte priradiť skutočnosť, že používateľ si môže vytvárať vlastné adresáre v rámci už existujúcej adresárovej štruktúry podľa svojho vlastného uváženia a podľa svojich potrieb. Táto vlastnosť sa v plnej miere využíva pri prístupe k pošte protokolom IMAP [6].

## 2.4 Efektívnosť využívania diskovej kapacity serverom Postfix

Na reálne zistenie ako efektívne využíva mailový server Postfix diskovú kapacitu som uskutočnil jednoduchý test. Na počítači typu PC s nainštalovaným OS Linux som vytvoril 10 nových testovacích používateľov posttest1 až posttest10. OS zabezpečoval iba služby nevyhnutné pre beh mailového servera a ostatné služby neboli spustené. V tomto stave som poslal všetkým testovacím používateľom mail obsahujúci 10MB textových ASCII znakov. V rámci dostupných nástrojov súborového systému som porovnal využitie kapacity súborového systému, kde sa ukladali maily pred testom a po teste. Rozdiel o veľkosti 100MB zodpovedal, došlým 10 emailom, každý o veľkosti málo väčšej než 10MB. Pri podrobnejšej analýze som určil skutočnosť, že všetky emaily sú identické až na pár riadkov v hlavičke mailu. Na obrázkoch 7 a 8 sú hlavičky emailov 2 používateľov, rozdiely sú podčiarknuté.

```
Return-Path: <root@vlk.ynet.sk>
X-Original-To: posttest1@vlk.ynet.sk
Delivered-To: posttest1@vlk.ynet.sk
Received: by vlk.ynet.sk (Postfix, from userid 0)
      id C936D8CD9D; Mon, 11 Nov 2006 15:16:23 +0100 (CET)
Received: from localhost (localhost [127.0.0.1])
      by vlk.ynet.sk (Postfix) with ESMTTP id C7BF62C2A;
      Mon, 11 Dec 2006 15:16:23 +0100 (CET)
Date: Mon, 11 Nov 2006 15:16:23 +0100 (CET)
From: root <root@vlk.ynet.sk>
To: undisclosed-recipients: ;
...
```

Obrázok 7: Začiatok hlavičky emailu doručeného používateľovi posttest1

```
Return-Path: <root@vlk.ynet.sk>
X-Original-To: posttest2@vlk.ynet.sk
Delivered-To: posttest2@vlk.ynet.sk
Received: by vlk.ynet.sk (Postfix, from userid 0)
      id C936D8CD9D; Mon, 11 Nov 2006 15:16:23 +0100 (CET)
Received: from localhost (localhost [127.0.0.1])
      by vlk.ynet.sk (Postfix) with ESMTTP id C7BF62C2A;
      Mon, 11 Dec 2006 15:16:23 +0100 (CET)
Date: Mon, 11 Nov 2006 15:16:23 +0100 (CET)
From: root <root@vlk.ynet.sk>
To: undisclosed-recipients: ;
...
```

Obrázok 8: Začiatok hlavičky emailu doručeného používateľovi posttest2

Pri teste bol použitý jeden identický obsah emailu, ktorý sa doručil fyzicky do mailového adresára každému používateľovi. Z pohľadu využitia diskovej kapacity je to zbytočné plytvanie miestom, nakoľko sa dva prípadne viac súborov s rovnakým obsahom ukladá viacnásobne na súborový systém.



## 2.5 Špecifikácia požiadaviek

Požiadavky na operačný systém:

- ♦ kompatibilný s emailovým serverom Postfix

Požiadavky na navrhovaný optimalizačný modul:

- ♦ spoľahlivosť – každý email, ktorý dostane modul na spracovanie, úspešne spracovaný, v prípade chybového stavu bude tento stav oznámený predchádzajúcemu modulu, respektíve bude v prípade fatálnej chyby vygenerovaný email odosielateľovi o príčine nedoručenia emailu
- ♦ bezpečnosť – modul neumožní žiadny neoprávnený prístup k emailovej komunikácii používateľom, ktorí nemajú na túto činnosť oprávnenie
- ♦ efektívnosť využitia diskovej kapacity – hlavný dôraz je kladený na túto vlastnosť modulu
- ♦ flexibilita – každý server pracuje v iných podmienkach, ktorým je potrebné sa prispôbiť, čo zabezpečia konfiguračné direktívy
- ♦ otvorený zdrojový kód – zabezpečí ďalší možný rozvoj modulu pre potreby možno aj ďalších emailových serverov

## 3 Návrh riešenia

### 3.1 Možné spôsoby riešenia

Vyššie popísaný test poukázal na problém neefektívneho obsadzovania diskovej kapacity emailovým serverom Postfix. Na riešenie uvedeného problému je vhodné využiť vlastnosť používaných súborových systémov, ktoré umožňujú pristupovať k jednému fyzicky uloženému súboru aj z viacerých miest v súborovom systéme. Konkrétnou a užitočnou vlastnosťou sú linky. Hlavná myšlienka riešenia: pri lokálnom doručovaní zistiť, či už daný mail nebol v určitom časovom rámci doručený a ak bol, aktuálnemu adresátovi vytvoriť v mailovom adresári iba link na už raz uložený mail.

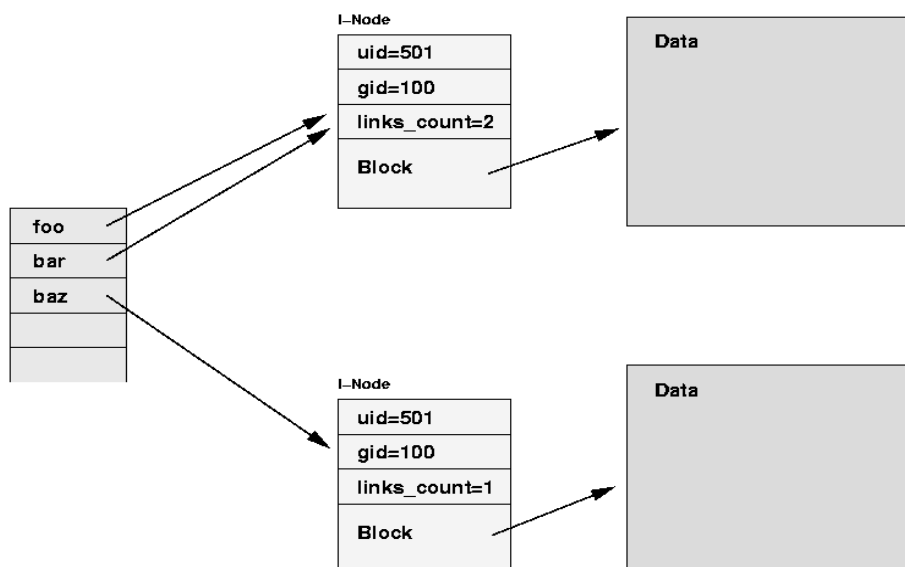
### 3.2 Linky - hardlink a symlink, iné riešenia

Rozoznávame dva druhy linkov. Hardlinky a symlinky. Každý druh ma svoje výhody aj nevýhody.

*Hardlink* je označenie situácie kedy jedno číslo inodu je odkazované viacerými menami súboru. Tieto mená sa môžu nachádzať v tom istom adresári, alebo v rôznych adresároch, ale musia sa nachádzať na tom istom súborovom systéme. Pri takomto usporiadaní sa zmena obsahu súboru alebo jeho atribútov (prístupové práva, čas, veľkosť, vlastník, ... ) prejaví na všetkých miestach, kde je dané číslo inodu odkazované. Prístupovanie k súboru cez ľubovoľné meno je rovnocenné. Výhodou hardlinkov je skutočnosť, že priamo v inode na odkazovaný súbor existuje počítadlo (reference counter), ktorý zaznamenáva počet existujúcich hardlinkov na daný súbor.

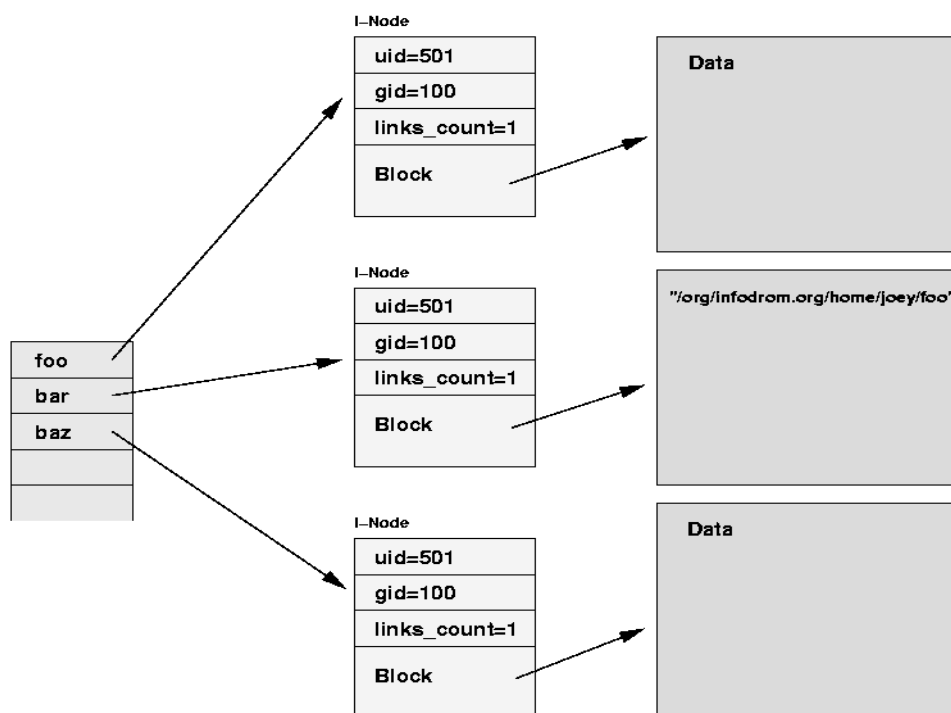
*Symlink* (tiež nazývaný softlink) je situácia, kedy adresár miesto toho, aby názvu súboru priradil číslo inodu, priradzuje mu meno iného súboru. Výhodou symlinku (v porovnaní s hardlinkom) je to, že môže prekračovať hranice súborového systému. Nevýhodou je to, že odkazovaný súbor nenesie žiadnu informáciu o tom, že naň niekto odkazuje. Ak je cieľový súbor zmazaný, dostávame neplatný symlink. Prístupové práva sa overujú vzhľadom na cieľový súbor.

Praktický prípad linkov aj s hodnotami *reference counter* (prípadne sa nazývajú *links\_count*) znázorňujú obrázky 9 a 10.



Obrázok 9: Princíp hardlinkov

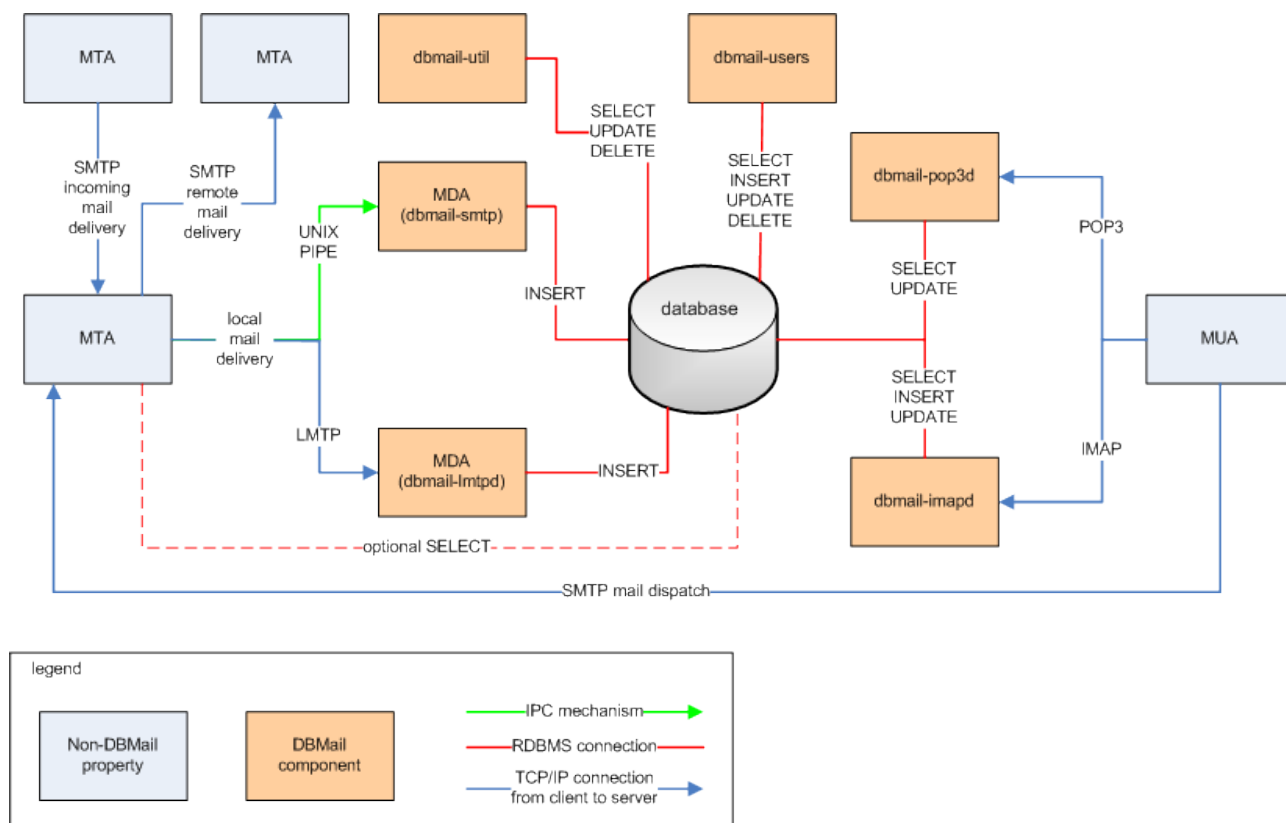
### 3.2 Linky - hardlink a symlink, iné riešenia



Obrázok 10: Princíp symlinkov

Ďalší možný spôsob riešenia efektívneho využívania diskového priestoru je využitie databázy ako úložiska emailov. V tomto prípade sú v databáze uložené a prepojené tabuľky, do ktorých sa ukladajú priamo emaily. Identické emaily sú uložené v databáze iba raz a právnické vzťahy sú riešené na úrovni vlastníckych práv k záznamu alebo pomocou relácií tabuliek. Výhodou je škálovateľnosť, manažovateľnosť, rýchlosť, flexibilita. Nevýhoda je v samotnej databáze, ktorá je ďalším elementom v celom procese a v porovnaní s ukladaním do súboru sa pri práci s databázou spotrebuje väčšie množstvo výpočtového výkonu. Tento výpočtový výkon je potrebný na databázový manažment. Použitím databázy sa celý emailový systém stane zložitejším a aj náchylnejším na možné chyby. Napriek spomenutým nevýhodám sa databáza využíva aj v praxi. Samotný emailový server *Microsoft Exchange* využíva databázu na odkladanie emailov. Z nekomerčných verzií je najznámejším balík programov pod menom *DBMail*, ktorého štruktúra je zobrazená na obrázku 11. Nespornou nevýhodou koncepcie použitia databázy je nutnosť použitia osobitných démonov služieb (IMAP, POP3), ktorí vedia spolupracovať s emailami v databáze. Bežne používaní démoni nemajú zvyčajne implementované databázové funkcie a tak balík *DBMail* obsahuje vlastné implementácie imap a pop3d démonov.

### 3.2 Linky - hardlink a symlink, iné riešenia



Obrázok 11: Schéma a koncepcia DBMail

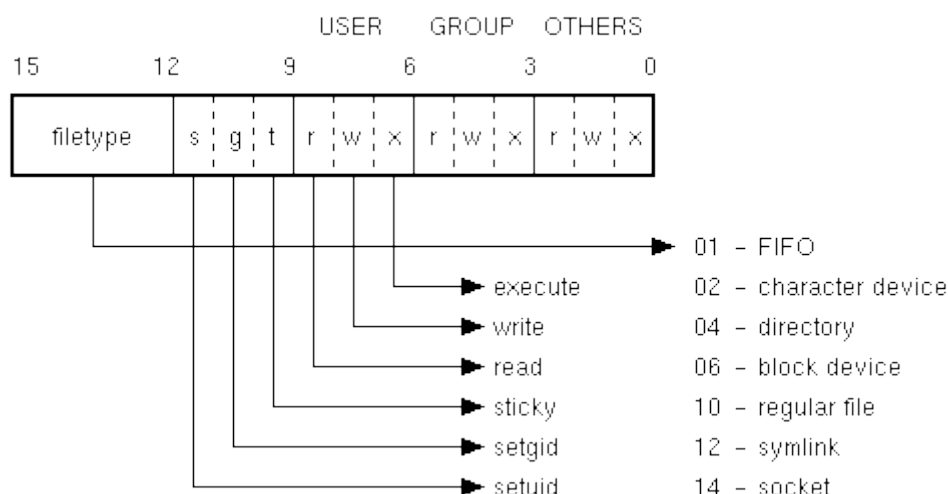
### 3.3 Prístupové práva

Jeden mail, môže byť doručený niekoľkým používateľom. Napríklad jedna pracovná skupina zamestnancov dostane mailom pracovnú úlohu. Navrhovaný mailový systém ošetrí vzniknutú situáciu uložením jednej kópie mailu prvému adresátovi a u ostatných adresátoch detekuje existenciu dotyčného mailu a v mailových adresároch ostatných adresátov vytvorí iba linku na už uložený email. Problém nastane s akými právami uložiť email resp. linku. Problém linky je vyriešený nakoľko ona samotná ukazuje na mail, čiže práva sa nevzťahujú na linku, ale priamo na linkovaný súbor – email.

Triviálnym riešením je využitie štandardných bitových práv. (vlastník, skupina, ostatní - čítanie/zápis/vykonávanie). Význam jednotlivých bitov a členenie práv je znázornený na obrázku 12. Nevýhodou tohto riešenia, je situácia, keď všetci adresáti emailu sú používatelia z jednej skupiny avšak existujú aj ďalší členovia tejto skupiny, ktorý tento súbor (email) nedostali a tak im prístupové práva neprináležia. Správne nastavenie bitových práv by v tomto prípade mohli byť vyriešené vytvorením 2 nových skupín

### 3.3 Prístupové práva

(skupina, ktorá dostala email a skupina, ktorá nedostala email) a následným správnym priradením jednotlivých členov pracovnej skupiny do novovzniknutých skupín. Na túto akciu sú však potrebné práva administrátora systému a je zrejmé z tohto prístupu neustále vznikanie nových skupín. Počet skupín v systéme je stanovený OS a bežne sa pre 32 bitové OS pohybuje na úrovni 65 536, čo je pre veľké emailové servery nepostačujúce.



Obrázok 12: Bitové práva a význam jednotlivých bitov

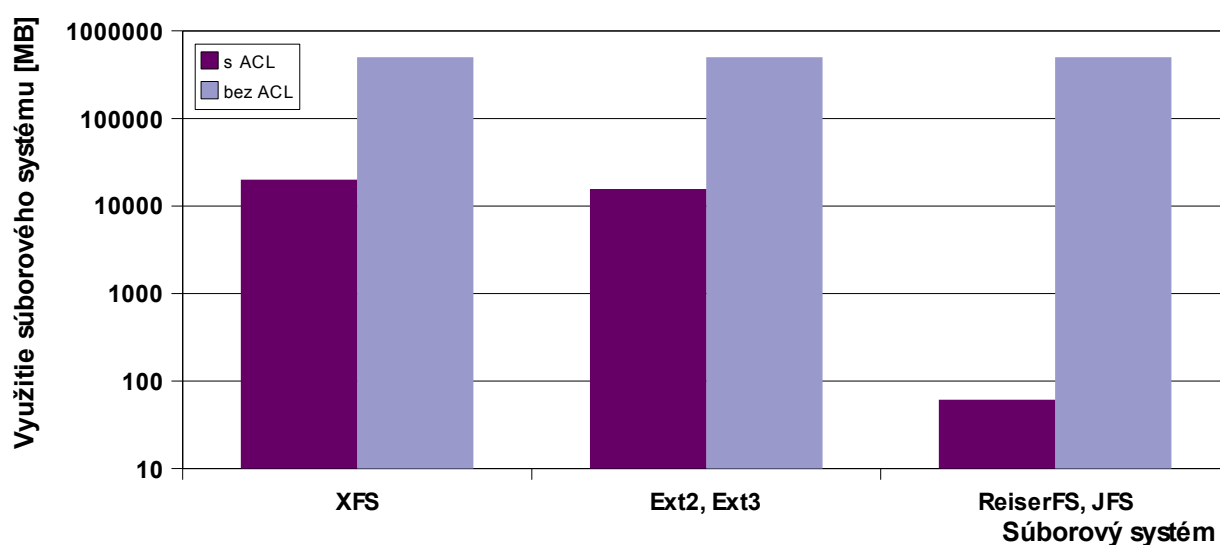
Aby sa zbytočne nevytváral veľký počet skupín, len na zabezpečenie prístupových práv k emailom, je namieste použitie POSIX ACL (Acces Control List) [7] [8]. ACL umožňujú vymenovať prístupové práva pre jednotlivých používateľov ku konkrétnemu súboru alebo adresáru. Obmedzenia na počet ACL záznamov sú závislé na použitom type súborového systému ako ukazuje tabuľka 3. Nedostatkom použitia ACL je aj zvýšenie prístupovej doby. Porovnávací tabuľka prístupových časov u rôznych súborových systémov s použitím a bez použitia ACL je k dispozícii v [prílohe A](#).

Súborový systém	Max. počet ACL záznamov
XFS	25
Ext2, Ext3	32 (môže byť aj viac – určené veľkosťou bloku [9])
ReiserFS, JFS	8191

Tabuľka 3: Maximálny počet ACL záznamov v závislosti od súborového systému

### 3.3 Prístupové práva

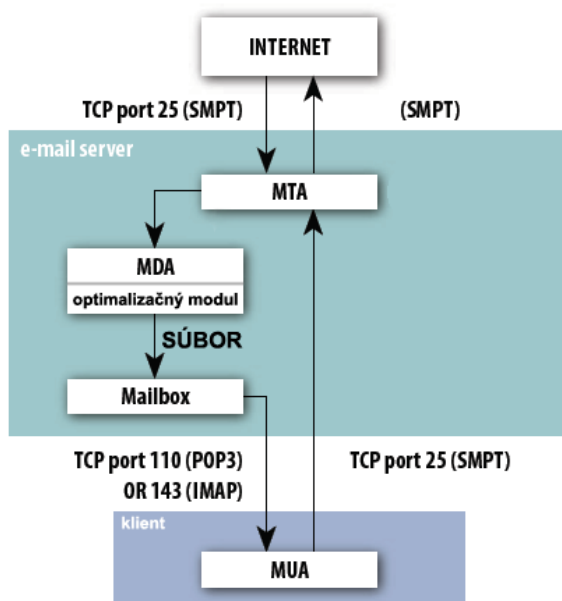
Všeobecné obmedzenie na počet ACL záznamov na súbor je možné jednoducho obísť uložením každého x-tého emailu, kde x je rovné limitu súborového systému na počet ACL+1. Následné došlé identické emaily sa budú linkovať na naposledy uložený email. Aj pri súborovom systéme s najmenším počtom možných uložených ACL záznamov (XFS) sa ušetrí diskový priestor o veľkosti 24 krát veľkosť dotyčného emailu, čo predstavuje značné ušetrenie priestoru. Grafickú reprezentáciu ušetrenej diskovej kapacity pri jednotlivých súborových systémoch za predpokladu 100 000 identických 5MB emailov, pričom veľkosť linkov vzhľadom na veľkosť emailu zanedbávame, je znázornené na obrázku 13.



Obrázok 13: Využitie súborového systému pri optimalizovanom ukladaní emailov s použitím a bez použitia ACL

### 3.4 Integrácia s emailovým serverom Postfix

Postfix zabezpečujúci doručenie emailu do používateľského mailového adresára, nie je navrhnutý na optimalizáciu emailovej komunikácie pomocou linkov a tak je nutné optimalizačný modul umiestniť na koniec spracovávania, tesne za MDA, ako je znázornené na obrázku 14. Pri integrácii pred MDA by nastal problém ako sa má správať MDA, keď dostane duplikát už existujúceho emailu a taktiež ako mu túto skutočnosť oznámiť.



Obrázok 14: Integrácia optimalizačného modulu do Postfixu

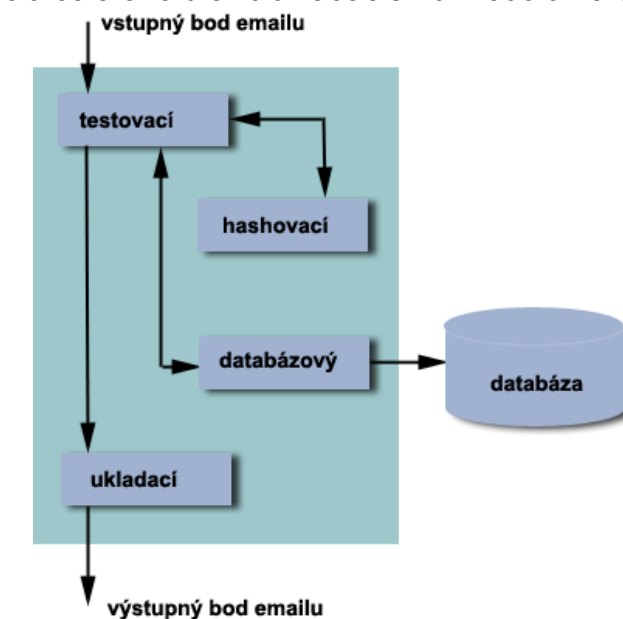
Po analýze problému navrhujem riešenie s nasledovnými požiadavkami:

- ♦ operačný systém kompatibilný s Postfix
- ♦ úložisko mailov maildir
- ♦ súborový systém s implementovaným ACL - ext2/3, ReiserFS, XFS, JFS

Operačný systém je vybraný v závislosti od požiadaviek Postfixu, na ktorý budem vytvárať optimalizačný modul. Maildir je požadovaný vzhľadom k tomu, že na samostatný email v mailbox-e nie je možné aplikovať osobitné ACL práva rozdielne od ostatných uložených emailov. Z analýzy vyplýva aj nutnosť ACL pre správne zabezpečenie prístupových práv jednotlivým používateľom, nakoľko bitové práva majú nevhodné obmedzenia, spomenuté vyššie pri analýze problémovej oblasti.

### 3.5 Vnútorne členenie optimalizačného modulu

Nasledujú obrázok 15 zachytáva navrhované vnútorné členenie optimalizačného modulu. Testovací modul prijme zo vstupu email a na jeho časť (telo) zavolá hashovaciu funkciu. Informácie potrebné na zistenie duplicitnosti emailu (hash, umiestnenie, časový údaj, ...) sa budú uchovávať v databáze. Navrhnutá štruktúra tabuľky je popísaná v tabuľke 4. Databáza má oproti ukladaniu údajov do súboru výhodu predovšetkým v rýchlom prehľadávaní údajov, čo bude mať priamy vplyv na rýchlosť rozhodovania a celkového času spracovania emailu celým modulom. V súčasnosti dokáže spolupracovať Postfix s viacerými typmi databáz. V záujme zachovania maximálnej flexibility bude obsahovať aj navrhovaný modul databázové rozhranie pre bežne používané databázy ako MySQL, PostgreSQL a iné. Podľa získaných hodnôt sa testovací modul rozhodne o uložení emailu do štandardného súborového tvaru v maildire alebo uložení emailu iba ako linku. Následne posunie informáciu čo a ako uložiť ukladaciemu modulu na uloženie.



Obrázok 15: Návrh vnútorné členenie optimalizačného modulu

Pri navrhovanom optimalizovanom linkovanom ukladaní emailov vzniká problém s informáciami, ktoré používateľ môže mať prístupné, aj keď by k nim nemal mať žiaden prístup. Z obrázkov 7 a 8 je vidieť, že v prípade linkovania emailu používateľa posttest2 na email používateľa posttest1, tento vidí v hlavičke X-Original-To a Delivered-To s údajmi, ktoré neprináležia jeho osobe. Z pohľadu používateľa sú tieto informácie nevýznamné a využívajú sa iba pri doručovaní emailu. Z tohto dôvodu je ich možné vymazať, o čo sa bude starať už funkcia, ktorá bude prijímať email.

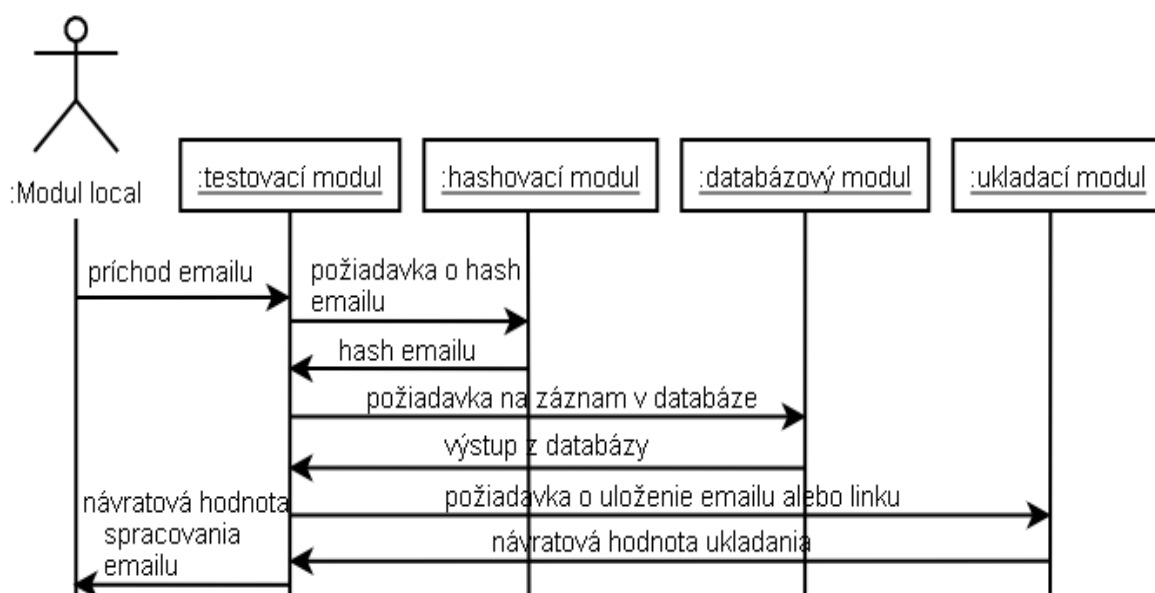


### 3.5 Vnútorne členenie optimalizačného modulu

Položka tabuľky	Poznámka
<b>email_id</b>	<b>primárny kľúč tabuľky</b>
<b>body_hash</b>	údaj o hash hodnote emailu
<b>inode</b>	číslo inodu uloženého emailu
<b>timestamp</b>	časová pečiatka príchodu emailu

Tabuľka 4: Štruktúra tabuľky pre záznam údajov pre optimalizačný modul

Obrázok 16 predstavuje sekvenciu postupností pri typickom spracovaní emailu navrhovaným optimalizačným modulom.



Obrázok 16: Sekvenčný diagram navrhovaného optimalizačného modulu

## 4 Technická dokumentácia

### 4.1 Implementácia optimalizačného modulu

Optimalizačný modul je implementovaný v jazyku C, nakoľko väčšina softvéru pod OS Linux je napísaná v tomto jazyku resp. C++.

Základná štruktúra, s ktorou pracujú jednotlivé funkcie implementácie je email. Podľa charakteru pracujú jednotlivé funkcie s príslušnými položkami nasledovnej štruktúry:

```
typedef struct {
    char *head;           //hlavička emailu
    char *body;           //telo emailu
    char *hash;           //hash tela emailu
    char *to;             //meno lokálneho užívateľa
    char *homedir;        //domovský adresár
    char *filepath;       //cesta spolu s názvom email súboru
    int to_uid;           //uid lokálneho užívateľa
    int done;             //flag spracovania
    long id;              //id záznamu v databáze
    long inode;           //číslo inodu
} email;
```

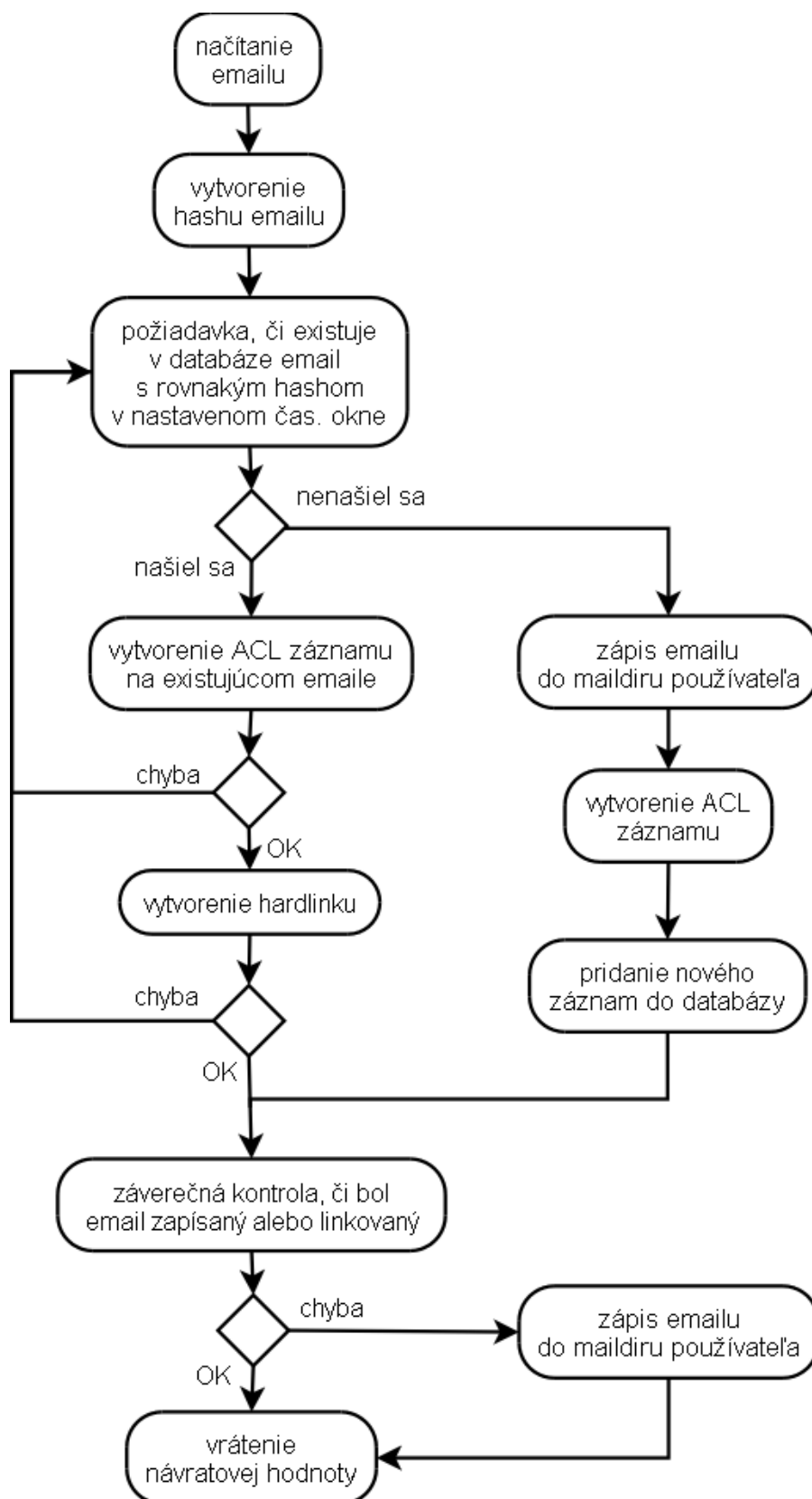
Ďalšie použité štruktúry:

- config – konfiguračné údaje
- SHA256\_CTX – štruktúra využívaná v hashovacej funkcii

Ako bolo spomenuté, na identifikáciu emailu sa využíva hash tela emailu. Existuje niekoľko druhov hashovacích algoritmov, ktoré prichádzali do úvahy na použitie. K najznámejším patrí nepochybne MD5. Nakoľko však existujú algoritmy, ktoré dokážu generovať kolízie pre MD5 v čase rádovo niekoľkých minút [20], považoval som použitie tohto algoritmu za nevhodné z bezpečnostného hľadiska. Obdobný problém nastáva pri použití SHA-1 [21]. Nakoniec s veľkým dôrazom na bezpečnostné hľadisko bol vybraný hashovací algoritmus SHA256, nakoľko v súčasnosti neexistujú reálne spôsoby, ktorými by bolo možné preň vytvoriť kolízie. Jeho výhodou je aj skutočnosť, že je schválený ako národný štandard hashovacej funkcie v USA (schválený *American National Standards Institute*) a v ďalších krajinách.

V požiadavkách na databázovú vrstvu sa nachádza aj nezávislosť na použitej databáze. Požiadavka je splnená použitím knižnice libdbi, ktorá umožňuje relatívne jednoduchou zmenou v konfiguračnom súbore zmeniť typ databázy.

## 4.2 Diagram činností



Obrázok 17: Rámcový diagram činností

## 4.3 Popis činnosti

Nasledovný popis detailne popisuje činnosť a akcie aplikácie. Popis sa odvíja od rámcového náčrtu činností aplikácie – obrázok číslo 17.

Samotná aplikácia je navrhnutá ako služba - démon, ktorý počúva na preddefinovanom porte. Služba je na začiatku spúšťaná s rootovskými právami, ktorých sa hneď na začiatku zbaví z bezpečnostných dôvodov. Pri práci s veľkými právami by mohla byť aplikácia reálne zneužitá na nechcené preniknutie do systému s danými právami. Po štarte sa nastaví práva na používateľa postfix a s jeho právami sa pracuje až do ukončenia, aby sa zamedzilo a minimalizovalo riziko zneužitia chýb aplikácie.

Po štarte a nastavení práv je načítaná konfigurácia z konfiguračného súboru, je inicializovaný systémový log pre našu aplikáciu a inicializované načúvanie na požiadavky na zadanom porte. Démon po príchode novej požiadavky vykoná akceptáciu požiadavky a vytvorí pomocou volania fork svojho potomka, ktorý obsluhuje požiadavku, pokiaľ on môže čakať na ďalšie požiadavky.

Prvou úlohou potomka je správne načítať a inicializovať premennú *new\_email*, ktorá obsahuje všetky potrebné údaje o emailoch. *New\_email* je vytvorený pomocou funkcie *readmail*, ktorá zabezpečí načítanie prichádzajúceho emailu, jeho rozdelenie na hlavičku a telo, určenie adresáta z hlavičky, jeho domovského adresára a jeho UID v systéme. Taktiež premenná štruktúry *new\_email->done* je inicializovaná na hodnotu 1, ktorá označuje ešte nespracovaný email.

V ďalšom kroku sa aplikácia pokúsi pripojiť na databázu (*connect\_db*) a vytvorí hash z tela emailu (*hash\_text*). V prípade chyby aspoň jednej tejto operácie aplikácia vykoná iba záverečné testovanie na spracovanie emailu (prem. *done*), kde zistí neúspešné spracovanie a uloží email adresátovi ako súbor do jeho maildiru. V prípade úspechu sa vygeneruje požiadavka na databázu o určenie identického emailu (*select\_by\_hash*) podľa konfigurácie (*time\_window*).

V prípade NULL odpovede (neexistencia identického emailu) na dotaz do databázy nasleduje zápis emailu do súboru (*write\_email*), ktoré v prípade úspechu označí email za spracovaný (*done=0*). Následne je pridaný nový záznam do databázy (*insert\_email*). V prípade pozitívnej odpovede na *select\_by\_hash* je vrátená štruktúra *ident\_email* s údajmi o tom, kde je uložený identický email s najstarším možným dátumom ukladania - príchodu. Táto štruktúra sa využije pri vytváraní linku, ktorý je ďalším krokom.

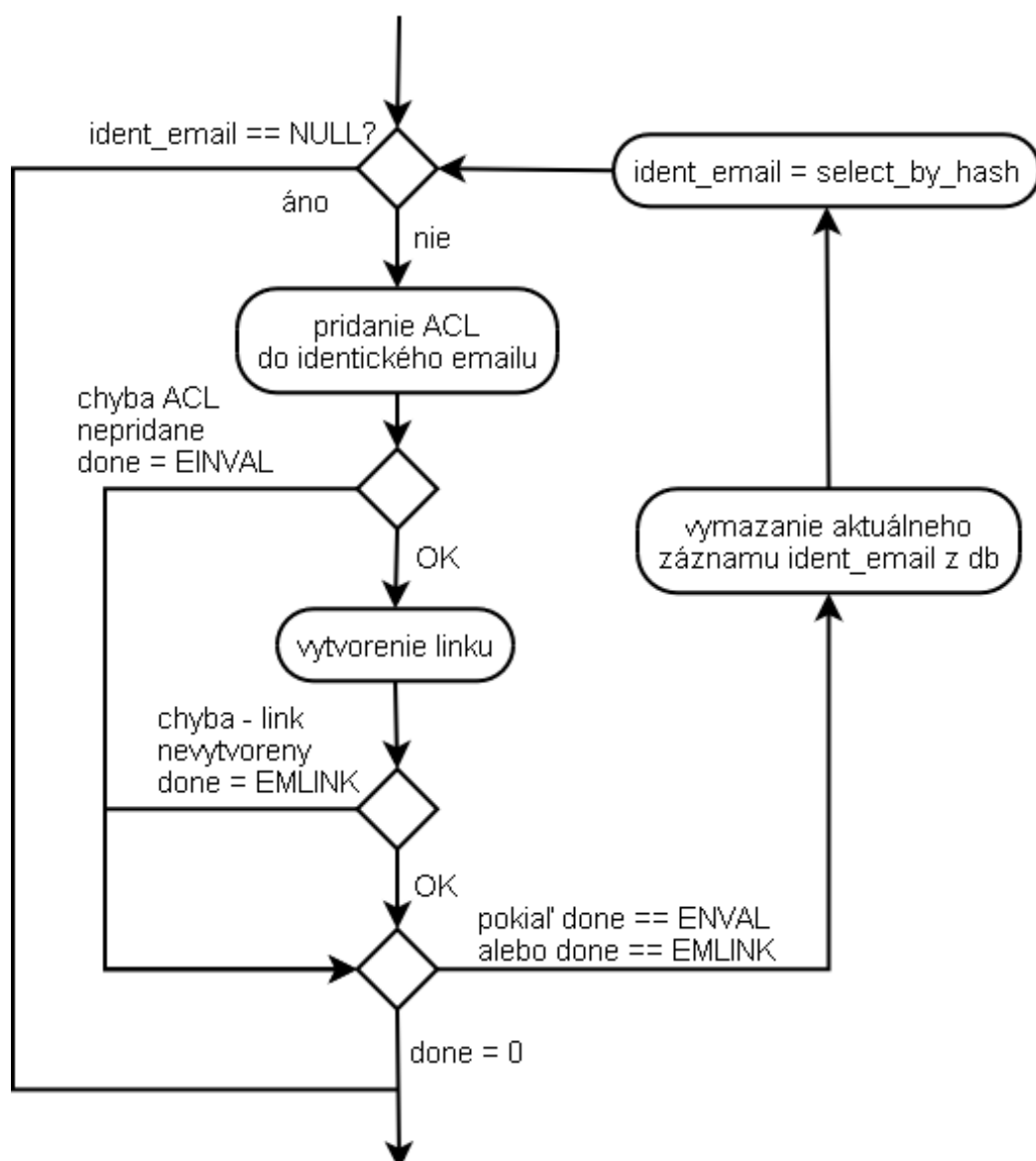
Linkovanie je realizované funkciou *link\_email*, ktorá v prvom rade skontroluje existenciu súboru daného názvu ako bol vrátený z databázy. Problém nastáva v prípade, že si používateľ už medzičasom prečítal email a tak tento email bol presunutý a adresára new (nové ešte neprečítané emaily) do adresára cur (prečítané emaily v inboxe). Problém sa ešte skomplikuje, keďže sa zmení nielen umiestnenie, ale aj názov emailu (názov súboru zmení pri prístupe imap alebo pop3 démon).

Riešenie tejto komplikácie spočíva v nájdení emailu v adresári cur. Adresár sa neprehľadáva na meno súboru, ale na číslo inodu, ktoré sa taktiež zaznamenáva v databáze. Po nájdení zhodného inodu sa určí hash tela emailu, aby bolo zaručené linkovanie správneho email. Príslušný názov súboru sa priradí nanamiesto už neaktuálneho názvu v *ident\_email*. Následne prebehne príprava na linkovanie. Príprava spočíva vo vytvorení ACL záznamu na existujúcom identickom emaily. Po tomto nastavení sa vytvorí hard-linka v maildir-e používateľa, ktorému aktuálne bol doručený email, pričom táto linka ukazuje na už uložený email v maildire, iného používateľa. V prípade úspešného linkovania sa nastaví *done* na 0. V prípade chyby linkovanie EMLINK (dosiahnutý maximálny počet hard-liniek je táto chyba priradená do premennej emailu *done*. Obdobným spôsobom je ošetrené na chybu maximálneho počtu ACL záznamov aj pridanie nového ACL záznamu.

Potomok, ktorý volal funkciu po linkovaní skontroluje *done* na chyby maximálnych počtov ACL a hardliniek. V prípade výskytu týchto chýb zavolá funkciu, ktorá vymaže príslušný záznam o identickom emaily z databázy (*delete\_email*) a zopakuje cyklus od požiadavky *select\_by\_hash*, pokiaľ sa email úspešne nespracuje, alebo pokiaľ už neexistujú ďalšie identické emaily v databáze. Popisované ošetrenie chybových stavov pridávania ACL záznamov a linkovania je zachytené na obrázku číslo 18.

Pokiaľ sa v tomto hlavnom pracovnom cykle nespracuje správne email, prichádza na rad už vyššie spomínané konečné testovanie, ktoré zapíše email ako súbor. V konečnom testovaní je kontrolovaný flag spracovania – *done* na rovnosť 0. Pokiaľ *done* nie je rovný 0 ukladá sa email bežným spôsobom.

Pokiaľ počas spracovania emailu nenastane žiadna kritická chyba (napr. chyba vytvárania emailu v domovskom adresári používateľa), je uvoľnená použitá pamäť a vrátená návratová hodnota 0, inak 1 (chyba).



Obrázok 18: Ošetrenie chybových stavov pridávania ACL a linkovania vo funkcii `link_email`

## 4.4 Inštalácia a požiadavky

Optimalizačný modul je všeobecne spustiteľný na OS Linux, ktoré spĺňajú nasledovné požiadavky:

- jadro OS je skompilované s podporou súborových systémov so zapnutými "extended attributes" a „POSIX Access Control Lists“
- balíček programov coreutils musí byť skompilovaný s podporou ACL
- balíček acl musí byť nainštalovaný - t.j. je k dispozícii knižnica libacl a prislúchajúce hlavičkové súbory (napr. acl.h)
- súborové systémy, na ktoré sa budú ukladať emaily musia podporovať POSIX ACL a musia byť namountované s parametrom acl

Pre systém Gentoo, na ktorom bol vyvíjaný modul odporúčam dokumentáciu k ACL na:

[http://gentoo-wiki.com/HOWTO\\_Use\\_filesystem\\_ACLs](http://gentoo-wiki.com/HOWTO_Use_filesystem_ACLs).

Databázové funkcie využívajú funkcie knižnice libdbi, ktorá je pre správnu činnosť aplikácie taktiež nutná. Knižnica libdbi je dostupná na <http://libdbi.sourceforge.net/>. K libdbi prislúchajú príslušné ovládače podľa druhu použitej databázy. Ovládače k najrozšírenejším databázam (MySQL, PostgreSQL, Firebird/Interbase, SQLite/SQLite3) sú k dispozícii na:

<http://libdbi-drivers.sourceforge.net/>. Samozrejmosťou súčasťou systému je plne funkčná databáza, ktorá má podporu v libdbi-drivers (sql definíciu používanej tabuľky je možné nájsť v súbore *src/sql\_database.sql*).

Samotná inštalácia spočíva v skompilovaní zdrojových súborov do spustiteľnej binárnej formy. Na tento účel je vytvorený v adresári src súbor Makefile. Použitím nástroja make a tohto Makefile-u vznikne binárna spustiteľná aplikácia. Následné spustenie príkazu make s rootovskými právami a s parametrom install vykoná inštaláciu aplikácie do */usr/sbin* a konfiguračného súboru do */etc/postfix*.

Po inštalácii je nutné vykonať konfiguráciu. Konfiguračný súbor */etc/postfix/mailod.conf* obsahuje nasledovné položky:

- *database\_driver* – nastaviť podľa použitej databázy
- *database\_host* – nastaviť meno host-a databázy
- *database\_username* – prihlasovacie meno do databázy

## 4.4 Inštalácia a požiadavky

- `database_password` – prihlasovacie heslo do databázy
- `database_name` – názov databázy
- `time_window` – časový údaj určujúci počet minút, v ktorých aplikácia spätne od aktuálneho času hľadá identický email v databáze
- `debug` – logovacie výpisy sa vypisujú aj na `stderr`, určené predovšetkým na debugovacie účely
- `port` – číslo portu na, ktorom aplikácia počúva a očakáva emaily
- `backlog` – nastavuje koľko prichádzajúcich požiadaviek aplikácia prijme, pokiaľ je zamestnaná poslednou príchodskou požiadavkou (maximálna veľkosť queue čakajúcich spojení)

Po týchto potrebných nastaveniach je možné spustiť aplikáciu a priradiť ju k defaultne spúšťaným aplikáciám. Pre potreby distribúcie `gentoo` je distribuovaný init script v `etc/init.d/mailod`.

Posledným krokom je nastavenie emailového servera `Postfix` (prípadne iného), aby posielal emaily na nastavený port. V `/etc/postfix/main.cf` je potrebné editovať riadok s nastavením `mailbox_command` na nasledovné parametre:

```
mailbox_command = nc -q 0 database_host port
```

V tomto konkrétnom prípade slúži program `nc` na presmerovanie výstupu mailservera na port servera `database_host`. Parametre `database_host` a `port` sú určené v nastavení `/etc/postfix/mailod.conf`.

Je možné použiť aj iný druh presmerovania. Pre kontrolné a vývojárske účely odporúčam nastaviť aj logovanie všetkých dát do súboru pomocou nasledovného nastavenia:

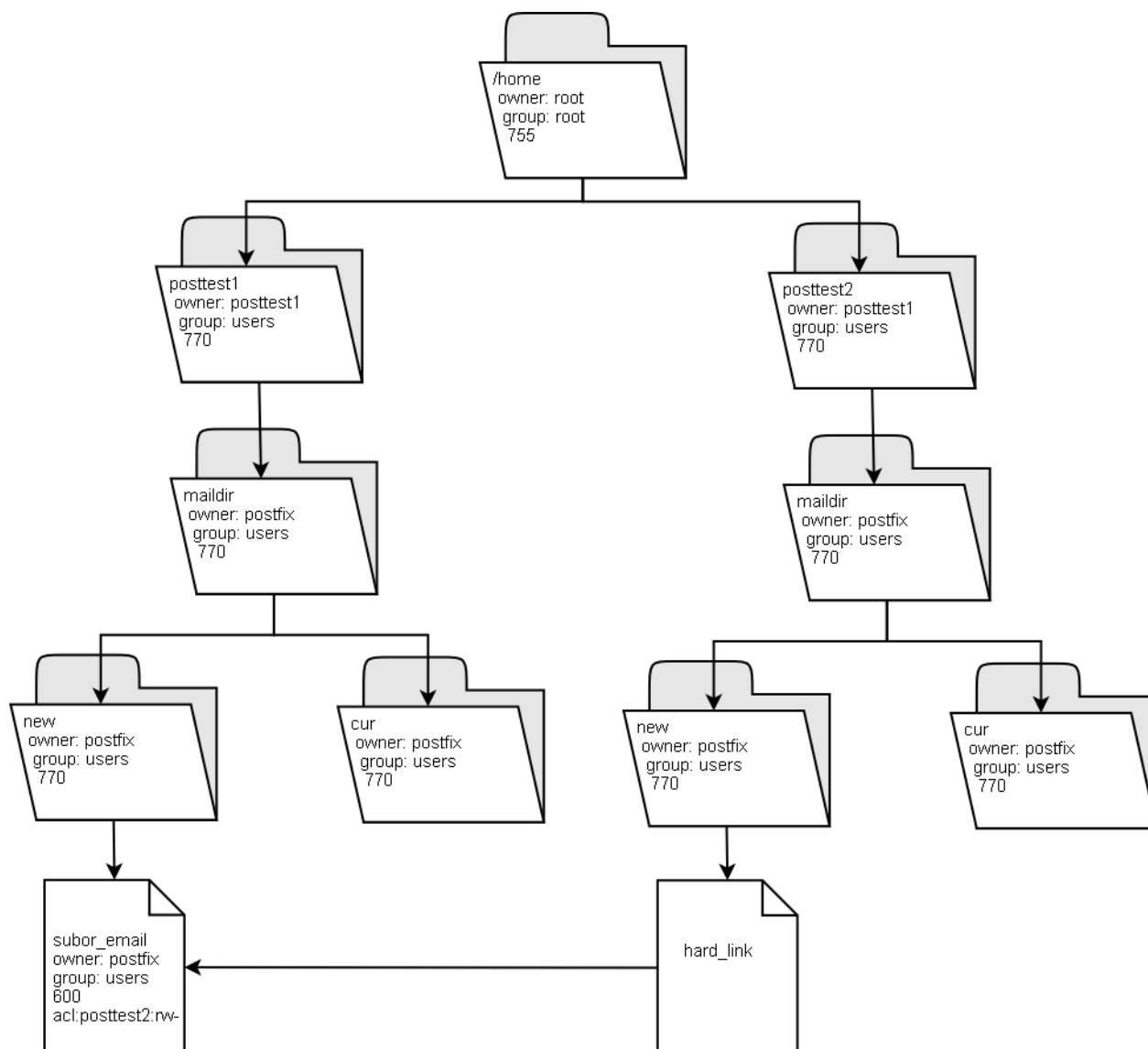
```
mailbox_command = tee -a /tmp/logg_of_redirect_data | nc -q 0 database_host port
```

Po následnom reštarte servera `Postfix`, už všetky emaily budú ukladané a optimalizované optimalizačným modulom.



## 4.5 Nastavenie práv a ACL

Nainštalovaný a sprevádzkovaný modul predpokladá nastavené práva podľa obrázka číslo 19, ktorý ilustruje situáciu, keď má používateľ `posttest2` nalinkovaný email od používateľa `posttest1`. Podotýkam, že adresáre `new` a `cur` majú nastavené aj defaultné ACL záznamy pre konkrétného používateľa `posttestx`, vďaka ktorým nie je potrebné pri zápise súboru (emailu) do týchto adresárov pridávať ACL záznamy týmto používateľom. O ich práva sa postará automaticky OS. Skript, ktorý zabezpečí nastavenie práv a vlastníctiev pre testovacích používateľov je archivovaný v `test/chowni_maildiry` (rieši aj práva pre `imap` démona). Uvedené nastavenie počíta iba s `imap` a `pop3` prístupom k emailom.



Obrázok 19: Nastavenie prístupových práv k adresárom a súborom

## 4.6 Overenie riešenia

V systéme boli vytvorený používatelia `posttest1` – `posttest10`, ktorý mali nastavené príslušné práva na svoje maildiry. Testovanie prebehlo z `localhost-u` odoslaním testovacích textov postupne všetkým `posttestx` používateľom príkazom `cat test_text | mail -s „testing mailod“ posttestx`. Vylistovanie obsahu maildir-ov používateľov preukázala správne nalinkovanie emailov. Emaily sa v tomto prípade uložili iba raz s príslušnými ACL záznamami. Úspora diskovej kapacity predstavuje takmer 90% oproti neoptimalizovanému ukladaniu. Podotýkam, že testovacia situácia bola ideálna a v reálnej prevádzke sú odlišné prevádzkové podmienky a stavy. Použitý testovací skript je v `test/sends_test_mails.sh`.

## 5 Zhodnotenie

Navrhnutý a implementovaný optimalizačný modul, optimalizuje využitie diskovej kapacity v rámci možností súborových systémov. Úplne ideálne riešene by bolo ukladanie emailu s rovnakou hash hodnotou tela iba raz na súborový systém. Vzhľadom k obmedzeniam a limitom každého súborového systému nie je možné dosiahnuť tohto stavu. Limitujúcim faktorom v riešenom probléme je obmedzený počet ACL záznamov na súbor, obmedzený počet hardliniek ukazujúcich na jeden súbor, neexistencia hardliniek medzi rôznymi súborovými systémami. Pri počtoch rovnakých emailov rádovo tisícky sa javí ako výhodnejšie použitie riešenia s databázou. Ovšem pre bežných používateľov s vlastným emailovým serverom, pre malé a stredné firmy, ktoré bežne nedostávajú tisícky zhodných emailov je vhodné využiť prezentované riešenie v tejto práci. Pre praktické nasadenie do reálnej prevádzky optimalizačného modulu by bolo vhodné doplniť súčasný emailový server Postfix o databázové funkcie na zápis a zaimplementovať napísané funkcie do jeho modulu local. Týmto krokom, by bolo možné efektívne riešiť problémy napríklad s alias-my, ktoré moja aplikácia nerieši.

## 6 Záver

Pre potreby ďalšieho rozvoja aplikácie sú dostupné všetky zdrojové kódy (Subversion repository), vrátane tejto práce na adrese: <http://code.google.com/p/mailod>. V závere svojej práce sa chcem poďakovať za myšlienky, podnety a nápadité výklady problematiky svojmu vedúcemu Ing. Jánovi Mátemu. Taktiež moja vďaka patrí Dinkovi Korunicovi [17] za jeho trpezlivosť pri mojich otázkach a za preklady odborných výrazov.

## 7 Slovník pojmov problémovej oblasti

*ACL* – Acces Control List

súčasť rozšírených atribútov súborového systému umožňuje definovať prístupové práva pre jednotlivých používateľov systému.

*DNS* – Domain Name System

Systém umožňujúci preklad doménových adries na IP adresy.

*IMAP* – Internet Message Access Protocol

Protokol na sprístupňujúci emaily v mailboxe. Špecifikuje ho RFC2060.

*inod*

dátová štruktúra používaná v súborovom systéme, ktorá popisuje jeden súbor. Nesie informácie o tom, aké sú prístupové práva k danému súboru, kedy došlo k poslednej zmene, k prístupu k súboru, kto je jeho vlastníkom, vlastníacou skupinou, počet hardliniek a predovšetkým informácie o tom, kde sa v súborovom systéme nachádza obsah súboru.

*ISP* – Internet Service Provider

Poskytovateľ internetového pripojenia.

*MDA* – Mail Delivery Agent

Program (maildrop, procmail, deliver, local.mail, atď.) doručujúci email do používateľského mailboxu.

*MTA* – Mail Transfer Agent

Program (inštancia Sendmail, Postfix, Qmail, Exim, atď.) prijímajúci emailové správy z lokálnej a/alebo vzdialených domén, ktoré potom preposiela ďalej. Všeobecne sa hovorí, že MTA je agent prenášajúci maily k ďalším MTA a/alebo MUA.

*MUA* – Mail User Agent

Program spracujúci emaily (MS Outlook, Mozilla Mail, Eudora, atď.) , ktorý používateľovi zjednodušuje prístup a spracovanie emailov (napr. čítanie, vytváranie, tlač, triedenie atď.).

**MAILBOX**

Súbor alebo súborová štruktúra na disku, slúžiaca na ukladanie emailov.

Poznámka: Existuje niekoľko formátov mailboxov v Linuxových OS.

**POP3 – Post Office Protocol Version 3**

*Protokol na výber emailov z mailboxu. Špecifikuje ho RFC1939.*

**SMTP – Simple Mail Transfer Protocol**

Protokol doručujúci a prenášajúci emaily. Špecifikuje ho RFC0821.

## 8 Zoznam použitej literatúry

- [1] KYLE D. DENT: Postfix: The Definitive Guide. Sebastopol: O'Reilly, 2003. 278 s., ISBN: 0-596-00212-2
- [2] CRAIG HUNT: sendmail Cookbook. Sebastopol: O'Reilly, 2004. 408 s., ISBN: 0-596-00471-0
- [3] ESET, spol. s r.o.: NOD32 for Linux Server. eset.sk, [ cit. 12. nov. 2006, 12:40 GMT ]. Dostupné na webovskej stránke (world wide web):  
<http://u4.eset.com/manuals/nod Ug linux en.pdf>
- [4] POSTEL, J.: SIMPLE MAIL TRANSFER PROTOCOL (RFC 821). ietf.org, 1982, [ cit. 12. nov. 2006, 12:40 GMT ]. Dostupné na webovskej stránke (world wide web):  
<http://www.ietf.org/rfc/rfc0821.txt>
- [5] MYERS, R.: Post Office Protocol - Version 3 (RFC 1939). ietf.org, 1996, [ cit. 12. nov. 2006, 12:40 GMT ]. Dostupné na webovskej stránke (world wide web):  
<http://www.ietf.org/rfc/rfc1939.txt>
- [6] CRISPIN, M.: INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1 (RFC 2060). ietf.org, 1996, [ cit. 12. nov. 2006, 12:40 GMT ]. Dostupné na webovskej stránke (world wide web):  
<http://www.ietf.org/rfc/rfc2060.txt>
- [7] EMERY, V.: Using ACLs with Fedora Core 2 (Linux Kernel 2.6.5). VanEmery.Com, [ cit. 12. nov. 2006, 12:40 GMT ]. Dostupné na webovskej stránke (world wide web):  
<http://www.vanemery.com/Linux/ACL/linux-acl.html>
- [8] GRÜNBACHER, A.: POSIX Access Control Lists on Linux. suse.de, 2003, [ cit. 12. nov. 2006, 12:40 GMT ]. Dostupné na webovskej stránke (world wide web):  
<http://www.suse.de/~agruen/acl/linux-acls/online/>
- [9] TUCHBREITER, J.: mailing list acl-devel, „Maximum number of acls on directory on ext3: 28? (bs does not matter, 1024/2048/4096 tested)“. bestbits.at, 2005, [ cit. 12. nov. 2006, 12:40 GMT ]. Dostupné na webovskej stránke (world wide web):  
<http://acl.bestbits.at/pipermail/acl-devel/2005-January/001829.html>
- [10] domovská stránka mailového servera *Postfix* (nov. 2006)  
<http://www.postfix.org>
- [11] domovská stránka *Sendmail* (nov. 2006)  
<http://www.sendmail.org/>

- [12] domovská stránka *Exim* (nov. 2006)  
<http://www.exim.org/>
- [13] REGENTS OF THE UNIVERSITY OF CALIFORNIA.: How Much Information? 2003. bestbits.at, 2003, [ cit. 12. nov. 2006, 12:40 GMT ]. Dostupné na webovskej stránke: <http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/internet.htm>
- [14] domovská stránka *Microsoft Exchange* (nov. 2006)  
<http://www.microsoft.com/exchange>
- [15] Internet Research Reports 2006 Mail (MX) Server Survey. securityspace.com, [ cit. 12. nov. 2006, 12:40 GMT ]. Dostupné na webovskej stránke (world wide web): [http://www.securityspace.com/s\\_survey/data/man.200610/mxsurvey.html](http://www.securityspace.com/s_survey/data/man.200610/mxsurvey.html)
- [16] KNOTEK, M.: prepis online diskusie Zabezpečení operačních systémů. microsoft.com, [ cit. 12. nov. 2006, 12:40 GMT ]. Dostupné na webovskej stránke (world wide web): <http://www.microsoft.com/cze/technet/chat/chats/chat20061004.mspix>
- [17] RAKO, KORUNIC, DOBRENIC: Testiranje performansi OpenSource SMTP poslužitelja. dkorunic.net, [ cit. 12. nov. 2006, 12:40 GMT ]. Dostupné na webovskej stránke (world wide web): <http://dkorunic.net/pdf2/MTA-testiranje.pdf>
- [18] MATTHIAS, A.: MTA Benchmark. uni-dortmund.de, [ cit. 12. nov. 2006, 12:40 GMT ]. Dostupné na webovskej stránke (world wide web): <http://www-dt.e-technik.uni-dortmund.de/~ma/postfix/bench2.html>
- [19] HP ProLiant DL360 G4 server benchmark results for Microsoft Exchange Server 2003. hp.com, [ cit. 12. nov. 2006, 12:40 GMT ]. Dostupné na webovskej stránke (world wide web): [http://h71019.www7.hp.com/ActiveAnswers/downloads/BenchmarkResults\\_Microsoft\\_Exchange2003onProLiantDL360G4\\_34proc.pdf](http://h71019.www7.hp.com/ActiveAnswers/downloads/BenchmarkResults_Microsoft_Exchange2003onProLiantDL360G4_34proc.pdf)
- [20] KLIMA, V.: Stránka věnovaná kolizím hašovacích funkcí. hyperlink.cz, [ cit. 12. feb. 2007, 12:40 GMT ]. Dostupné na webovskej stránke (world wide web): [http://cryptography.hyperlink.cz/2004/kolize\\_hash.htm](http://cryptography.hyperlink.cz/2004/kolize_hash.htm)
- [21] Schneier on Security. schneier.com, [ cit. 12. feb. 2007, 12:40 GMT ]. Dostupné na webovskej stránke (world wide web): [http://www.schneier.com/blog/archives/2005/02/cryptanalysis\\_o.html](http://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html)

## 9 Príloha A

Použitie ACL má aj svoju nevýhodu, a to síce zvýšenie prístupovej doby. Pre potreby výberu správneho súborového systému pripájam nasledovnú tabuľku s prehľadom prístupových časov s ACL a bez ACL na bežne používaných súborových systémoch.

Súborový systém	prístupová doba s ACL [ $\mu$ s]	prístupová doba s ACL [ $\mu$ s]
Ext2	9	1743
Ext3	10	3804
ReiserFS	9	6165
XFS-256	14	7531
XFS-512	14	14
JFS	13	13