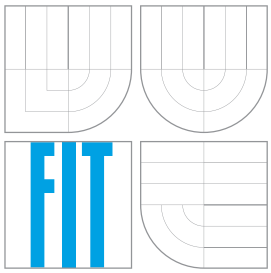
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

IMAP PROXY PRO POP3 POŠTU

IMAP PROXY FOR POP3 MAILBOXES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MIROSLAV KRUŽLIAK
AUTHOR

VEDOUCI PRÁCE — Doc. Dr. Ing. DUŠAN KOLÁŘ
SUPERVISOR

BRNO 2007

Zadání bakalářské práce/4609/2006/xkruzl00

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Kružliak Miroslav**
Obor: Informační technologie
Téma: **IMAP proxy pro POP3 poštu**
Kategorie: Počítačové sítě

Pokyny:

1. Prostudujte typy a možnosti IMAP serverů, jejich konfigurace a nastavení v prostředí Linux. Prostudujte možnosti stahování pošty z POP3 serverů a organizace pošty pro IMAP server. Dále prostudujte možnosti periodického spouštění procesů s proměnnou periodou. Studujte možnosti bezpečného uchování citlivých informací.
2. Navrhněte aplikaci pro automatické stahování pošty z řady POP3 servů a její doručení do jediného účtu v IMAP serveru, kde je autoamticky členěna do adresářů. Umožněte jednoduchou konfiguraci aplikace a bezpečné uložení hesel.
3. Implementujte tuto aplikaci a konfigurujte potřebné služby v prostředí Linux.
4. Ověřte činnost serveru na bezplatných e-amilových serverech.
5. Zhodnoťte přínos své práce, diskutujte možná využití, rozšíření a případné nedostatky své práce.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- První dva body zadání, pro bod 3 jen prototyp.

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 paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Kolář Dušan, doc. Dr. Ing.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2006
Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

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: **Miroslav Kružliak**
Id studenta: 84293
Bytem: Družstevná 323, 013 51 Predmier
Narozen: 05. 03. 1985, Žilina
(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: IMAP proxy pro POP3 poštu
Vedoucí/školitel VŠKP: Kolář Dušan, doc. Dr. Ing.
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 tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

....

Nabyvatel Autor

Abstrakt

Táto bakalárska práca sa zaoberá sťahovaním elektronickej pošty z POP3 serverov a jej následnou organizáciou na jediný účet IMAP servera. Taktiež študuje možnosti nastavenia a konfigurácie IMAP serverov v prostredí operačného systému Linux. Stručne sú tu porovnané protokoly IMAP a POP3 z implementačného hľadiska. Ďalej študuje možnosti periodického spúšťania procesov a bezpečného ukladania citlivých informácií. V tejto časti práce sú v skratke porovnané hlavné metódy šifrovania.

Kľúčové slová

IMAP server, POP3, elektronická pošta, cron, GPG, šifra.

Abstract

This bachelor's thesis deals with retrieving e-mails from different accounts on POP3 servers and their organisation in one account on IMAP server. It also studies settings and configuration of IMAP servers in enviroment of operating system Linux. Protocols IMAP and POP3 are briefly compared here from implementation point of view. Further it studies possibilities of periodical start of processes and secure saving of sensitive information. In this part of my thesis main methods of cryptography are shortly confronted.

Keywords

IMAP server, POP3, e-mail, cron, GPG, cipher.

Citácia

Miroslav Kružliak: IMAP proxy pre POP3 servery, bakalárska práca, Brno, FIT VUT v Brně, 2007

IMAP proxy pre POP3 servery

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Doc. Dr. Inq. Dušana Koláře.

.....

Miroslav Kružliak

14. máj 2007

Poďakovanie

Týmto by som chcel poďakovať Doc. Dr. Ing Dušanovi Kolářovi za jeho pomoc pri konzultáciach tejto bakalárskej práce.

Obsah

Kapitola 1

Úvod

Hlavnou témou tejto bakalárskej práce je štúdium spracovania elektronickej pošty ako aj implementácia aplikácie, ktorá je schopná spracovávať správy z rôznych serverov pracujúcich s protokolom POP3 a ich následné ukladanie na jediný účet IMAP servera.

Komunikácia prostredníctvom elektronickej pošty siaha historicky až pred zrod siete internet. Od tohto obdobia prešla komunikácia pomocou tejto služby radikálnymi zmenami. Avšak podstata, ktorou je sprostredkovanie komunikácie medzi ľuďmi, ostala nezmenená. Samotná komunikácia pomocou elektornickej pošty nie je interaktívna, ale jej obľúbenosť spočíva v spoľahlivosti a bezpečnej komunikácií medzi jej používateľmi. Za ďalšiu výhodu, prečo sa táto služba používa až do týchto dôb sa pokladá záruka bezpečného a nenáročného uloženia správ. V dobe, kedy si človek vytvára pre rôzne účely viacero poštových schránok, sa vynára problém prehľadnej správy poštových správ. Začala sa vytvárať potreba členiť elektronickú poštu do zložiek. Vznikajú tak rôzni klienti, ktorí sú schopní prehľadne uložiť správy svojim uživaťeľom. Táto bakalárska práca poukazuje na jednu z alternatív zefektívnenia prehľadnosti prijímanej pošty z rôznych zdrojov.

V tejto práci sa postupne zaoberám konfiguráciou a možnosťami nastavenia IMAP serverov v prostredí operačného systému Linux. Takisto sú tu diskutované jednotlivé druhy implementácií serverov z rôznych hľadísk, ako sú uloženie dát, uloženie citlivých informácií (hesiel) a autentizácia užívateľov. Ďalej sa tu zaoberám porovnávaním vybraných implementácií serverov s ohľadom na ich nastavenie a konfiguráciu.

S témou implementácie vyššie zmienenej aplikácie úzko súvisí aj téma bezpečného uloženia dát a periodicita spúšťania procesov v operačnom systéme Linux.

Bezpečnosť je jedna z najväčších priorít dnešnej informatickej spoločnosti. Sila počítačovej bezpečnosti spočíva hlavne v prevencii. Tú umožňuje vo veľkej miere šifrovanie. Veda, ktorá sa zaoberá šiframi a šifrovaním sa nazýva kryptografia. Prvé zmienky o tejto vede a jej využití siahajú do čias starovekého Grécka. Moderná história kryptografie sa rozvíjala paralelne s počítačovou technikou, ktorá jej zaručila veľký rozmach. V rámci bezpečnosti budú spomenuté pojmy ako symetrická, či asymetrická kryptografia, alebo štandard *OpenPGP*.

Taktiež bude diskutovaná otázka spúšťania procesov s premennou periódou, kde budú

z tohoto hľadiska nastolené dva prístupy, tj. štart procesu iným procesom a perióda spúšťania procesu kontrolovaná samotným procesom. Tieto dva prístupy budú porovnané a budú vyzdvihnuté ich výhody a nevýhody.

Súčasťou práce budú aj časti venované samotnej implementácií a návrhu už zmienenej aplikácie, ako i vhodnosť riešenia, možnosti vylepšenia a jej prínos. Tu je zahrnutý aj spôsob konfigurácie tejto aplikácie.

Kapitola 2

IMAP servery

Jednou z diskutovaných tém v tejto bakalárskej práci je aj práca a možnosti poštových serverov pracujúcich s protokolom IMAP. Postupne sa dopracujeme od krátkej histórie protokolu a jeho špecifikácie, až ku konkrétnemu porovnaniu a prehľadu uvažovaných serverov. Na konci kapitoly budú zhrnuté vlastné skúsenosti s inštaláciou a nastavením dvoch implementácií IMAP serverov.

2.1 História protokolu IMAP

IMAP protokol bol navrhnutý členom personálu Washingtonskej univerzity Markom Crispinom v roku 1986, ako odpoveď na rozsiahlo rozšírený POP protokol. Pôvodný protokol vystupoval pod názvom Interim Mail Access Protocol, ktorý bol neskôr nahradený protokolom s názvom Interactive Mail Access Protocol. Táto verzia bola definovaná v RFC[1] 1064 a bola prvou verziou vydanou pre verejné účely. Tento protokol bol označovaný ako IMAP2.

S príchodom MIME bola táto verzia protokolu upravená na IMAP2bis, ktorá bola podporovaná skoršími implementáciami poštového klienta Pine. Ďalším dôležitým míľnikom v histórií protokolu bolo založenie IMAP Working Group ako súčasť IETF[2] v roku 1990. Táto prevzala zdopovednosť nad vývojom protokolu IMAP a posunula jeho vývoj až do dnešnej podoby.

Aktuálna verzia protokolu bola vyvinutá v roku 1996, má názov IMAP4rev1 a je definovaná v RFC 3501.

Bližšie informácie nájdete v naledujúcej literatúre [8] a [7].

[1] Dokumenty, ktoré sú sériou memoránd zahrňujúcich nový výskum, inovácie a metodológie aplikovateľné na Internetové technológie.

[2] Dobrovoľnícka organizácia vyvíjajúca a zlepšujúca Internetové protokoly

2.2 Špecifikácia protokolu IMAP

Protokol IMAP je metóda vzdialenej práce s elektronickou poštou, ktorá je uchovávaná na poštových serveroch. Inými slovami, protokol slúži na vzdialený prístup programu poštového klienta k vzdialeným poštovým schránkam tak, ako keby boli na lokálnom počítači. Napríklad dovoľuje prácu s poštou na servere odkiaľkoľvek, bez ptoreby túto správu uložiť na lokálny disk.

Práca týchto serverov spočíva v sprostredkovaní a správou elektronickej pošty. Manipulácia servera s poštou je pre užívateľa zvyčajne skrytá, ten komunikuje len s poštovým klientom.

Jeho práca prebieha na aplikačnej vrstve referenčného modelu TCP/IP sieťovej komunikácie na porte 143. Bližšie Informácie o protokole nájdete v nasledujúcej literatúre: [1], [7].

2.3 Rozdelenie IMAP serverov

Cieľom tejto časti nie je zaoberať sa rozdelením serverov s ohľadom na operačný systém, na ktorom tieto servery pracujú, ale budú tu diskutované rozdiely v typoch použitých technológií popisujúcich jednotlivé riešenia implementácií a licencie, pod ktorými sú implementácie serverov vydávané. Väčšina serverov nie je jednoznačne určená len pre prácu s protokolom IMAP, ale aj pre prácu s ďalšími protokolmi umožňujúcimi komunikáciu pomocou elektronickej pošty, ako je POP3, či SMTP.

V nasledujúcich častiach predstavím rozdelenie serverov z hľadiska ich práce s dátami a z hľadiska licencií, pod ktorými sú jednotlivé implementácie vydávané.

2.3.1 Rozdelenie podľa licencie

- Open Source (GPL, LGPL)
- Súkromná licencia
- Špecifické licenčné podmienky, definované samotným výrobcom softvéru

Licencie označené ako *Open Source* sú sadou princípov, ktorá propaguje prístup k produkcií a návrhovému procesu rôznych tovarov, produktov, technickým správam alebo službám. Tento termín sa často spája so svetom informačných technológií, ale jeho použitie nájdeme aj v oblastich ako je vzdelanie, média a umenie. Viac sa dozviete tu [10].

Väčšina serverových implementácií vydávaných pod touto licenciou je voľne šíriteľná, pri niektorých sú však stanovené isté obmedzenia ako sú GPL a LGPL.

GPL licencia, pojednáva o zmene alebo doplnení zdrojového kódu programu, podmienkou je však vydať túto zmenu znova pod licenciou GPL.

LGPL licencia umožňuje používať zdrojové kódy vydané pod touto licenciou na vývoj softvéru, ktorého licenčné podmienky nie sú stanovené licenciou LGPL, to znamená, že takýto softvér môže byť komerčne predávaný.

Pod súkromnou licenciou je predávaný komerčný softvér, ktorý je súkromným vlastníctvom vlastníka tejto licencie. Vývoj tohoto softvéru je financovaný z predaja takýchto softvérových riešení.

2.3.2 Podľa spôsobu organizácie dát

- Databáza
- Súborový systém
- Vlastná

Spôsob spracovania dát do databáze spočíva v pridávaní týchto dát do štruktúr závislých na koncepcii samotnej databáze, server pristupuje k týmto dátam pomocou tzv. otázok. Práca nad databázou použitá v IMAP serveroch sa v zásade nelíši od koncepcií použitými pri komunikáci s databázovými servermi.

Servery používajúce súborový systém ukladajú užívateľské dáta do súborov konkrétneho operačného systému. Spôsob ukladania dát a prístupu ku nim závisí taktiež na operačnom systéme a súborovom systéme v ňom použitom.

Niektoré servery majú vlastnú koncepciu uloženia dát odlišnú od predchádzajúcich dvoch prístupov.

Jednotlivé implementácie môžu vo svojej konfigurácií umožňovať aj kombináciu týchto prístupov.

2.3.3 Podľa spôsobu uloženia autentifikačných údajov

- Databáza
- LDAP
- Súborový systém

Pri tomto rozdelení je situácia uloženia dát pomocou databáze a súborového systému podobná ako v rozdelení predchádzajúcom. Tu sa však musí manipulovať s dátami veľmi citlivo kvôli zamedzeniu prístupu tretích strán k prihlasovacím údajom užívateľov.

LDAP je protokol pre správu adresárov s informáciami o užívateľoch. Tento protokol pristupuje k dátam, ktoré sú uložené v stromovej štruktúre. Každý záznam v tejto štruktúre pozostáva zo sady atribútov. Bližšie o protokole sa môžete dozvediet napríklad tu [9].

Taktiež pri tomto rozdelení jednotlivé implementácie umožňujú použitie viacero zo spomínaných prístupov, poprípade rozšírenia uvedených prístupov. Veľa z implementácií používa k ukladaniu autentifikačných údajov kryptografické hashovacie algoritmy ako napr.: MD5.

2.3.4 Podľa implementovaných súčastí

Žiadny zo serverov sa nešpecializuje pre prácu nad jediným protokolom, ale pracuje nad celou radou súčastí. Toto rozdelenie je veľmi špecifické a záleží od distribúcie jednotlivých poštových serverov. Väčšina serverov pracuje okrem IMAP protokolu aj s protokolmi POP3 alebo SMTP. Taktiež veľa zo serverov podporuje kryptografické protokoly ako SSL alebo TLS, ktoré slúžia na zabezpečenú komunikáciu so medzi klientom a serverom.

Medzi ďalšie rozšírenia implementácií jednotlivých serverov patrí podpora NNTP[3] alebo Webmail[4].

2.3.5 Súhrn

V tejto časti som opisoval rozdelenie distribúcií jednotlivých serverov podľa zadania bakalárskej práce. Boli tu diskutované rozdelenie podľa rôznych kritérií. Viac o rozdelení s ohľadom na jednotlivé servery nájdete na týchto stránkach [12], ktoré slúžili aj ako základ tejto časti bakalárskej práce.

2.4 Vlastné skúsenosti s konfiguráciou a inštaláciou serverov

V tejto časti by som rád spomenul osobné skúsenosti s konfiguráciou dvoch poštový serverov Dovecot a Cyrus.

2.4.1 Porovnanie testovaných serverov

Server Dovecot je vyvinutý Timom Sirainenom a je publikovaný pod Open Source licenciou. Je orientovaný na jednoduchosť konfigurácie a administrácie servera.

Vývoj projektu servera Cyrus je zakorenený na Carnegie Mellon University taktiež pod licenciou Open Source. Tento vyvíjaný softvér je orientovaný na výkon poskytovaných služieb.

Poznatky z pohľadu jednoduchosti konfigurácie, nastavenia samotného servera, dostupnosti informácií, podporovaných protokolov a výkonu boli zhrnuté v nasledujúcej tabuľke.

2.4.2 Súhrn

Z pohľadu užívateľa je jedznoznačne server Dovecot prehľadnejšie a jednoduchšie riešenie, ktoré však zaostáva za vyšším výkonom servera Cyrus. Toto riešenie je jednoznačne orientované na vyššiu frekvenciu prenášaných dát a stabilitu, viď [14].

[3] Protokol slúžiaci na čítanie a uverejňovanie článkov v sieti Usenet.

[4] Rozšírenie implementácie serverov, ktorá umožňuje ich užívateľom prístup ku svojím kontám pomocou internetových prehliadačov.

Server	Výhody	Nevýhody
Cyrus	Výkon Jednoduchá inštalácia Dostupnosť zdrojov Pridávanie užívateľov Množstvo podporovaných protokolov Široké možnosti nastavení	Konfigurácia servera Nedostupnosť prehľadných informácií Zložitosť pridávania poštových schránok
Dovecot	Dostupnosť informačných zdrojov Prehľadnosť zdrojov Jednoduchosť konfigurácie Jednoduchá manipulácia Bezpečnosť Jednoduchá inštalácia	Slabši výkon

2.5 Zhrnutie

V tejto kapitole som sa venoval rozdeleniu IMAP serverov a štúdiu konfigurácie a možností dvoch implementácií týchto serverov. Poznatky získané zo začiatku tejto kapitoly som využil pri implementácií časti aplikácie komunikujúcej s IMAP serverom.

Kapitola 3

Spracovanie pošty pomocou protokolu POP3

Táto kapitola sa venuje problematike prístupu k elektronickej pošte prostredníctvom POP3 protokolu. Kapitola zahŕňa históriu samotného protokolu, špecifikáciu protokolu a diskutuje problémy vzniknuté pri implementácií klientov prcujúcich s protokolom POP3. Problémy vzniknuté pri implementácií aplikácií pracujúcimi nad týmto protokolom som sa rozhodol opisovať zároveň s porovnaním protokolov IMAP a POP3.

3.1 História protokolu POP3

História protokolu začala v skorých 80-tych rokoch, kde sa objavila potreba získavať a spracovávať elektronickú poštu priamo na počítači klienta. Ako odpoveď na túto potrebu bolo v roku 1984 publikované RFC 918. Ústrednou myšlienkou bolo sprostredkovať jednoduchým spôsobom získavanie elektronickej pošty na klientsky počítač. Pôvodný dokument obsahoval päť strán a protokol bol definovaný veľmi jednoducho.

V roku 1985 bol vydaný RFC 937, kde bol rozšírený pôvodný dokument a protokol bol pomenovaný POP2. Tento protokol je bohatší na sadu príkazov a odpovedí servera. Dôležitým rozšírením je schopnosť čítať určitú správu z poštovej schránky bez čítania všetkých uložených správ.

V roku 1988 bolo vydané RFC 1081, kde je opisaný protokol nesúci názov POP3. V tomto období sa dostávajú osobné počítače do pozornosti širokej verejnosti. POP3 protokol bol úzko spätý so svojím predchodcom protokolom POP2.

V 90-tych rokoch bolo vydaných niekoľko revízií spomínaného protokolu, ale pôvodny protokol z roku 1988 sa radikálne nezmenil. Od roku 1996, kedy bolo vydané RFC 1939, nebola vydaná už žiadna revízia protokolu POP3. Informácie sú čerpané z [6].

3.2 Špecifikácia protokolu

Protokol POP3 slúži na vzdialený prístup klienta k schránkam s elektornickou poštou. Užívateľovi umožňuje pracovať so svojou poštou v režime offline. To znamená, že klientská aplikácia je nútená organizovať poštu získanú z POP3 servera na strane klientského počítača. Protokol nepodporuje prácu so zložkami na strane servera a je možné pristupovať len do schránky prijatej pošty. Po stiahnutí správy, táto zvyčajne býva vymazaná. Tu nastáva problém prístupu k pošte z rôznych počítačov.

Tento protokol pracuje na aplikačnej vrstve referenčného modelu TCP/IP sieťovej komunikácie. Komunikácia pomocou tohto protokolu štandardne prebieha na porte 110. Viac o protokole sa dozviete tu [2].

3.3 Porovnanie protokolov IMAP a POP3

V tejto stati sú porovnané obidva už zmieňované protokoly z hľadiska jednoduchosti implementácie klienta a servera, sady príkazov jednotlivých protokolov a rozšírenosti na verejne dostupných poštových serveroch. Samotné porovnanie bolo pridané do tejto sekcie hlavne kvôli názornej ukážke rozdielnosti práce POP3 a IMAP protokolov, ako aj spôsob osvetlenia manipulácie s poštou pomocou protokolu POP3.

3.3.1 Spracovanie aplikácií pracujúcich nad protokolmi

Formálne sa implementácia klienta nad obidvoma protokolmi v zásade nelíši. Je treba však brať na zreteľ rôznorodosť odpovedí v jednotlivých implementáciach IMAP serverov. Klient pracujúci s protokolom IMAP však používa väčšiu množinu príkazov a musí sa vysporiadať aj s väčším množstvom odpovedí na tieto príkazy.

Na druhej strene, implementácie POP3 klientov sa musia vysporiadať s nedostatkami, ktorými disponuje protokol POP3 tak, aby bola aplikácia čo najviac užívateľsky prívetivá. V samotnej špecifikácií POP3 protokolu chýbajú príznaky správ, takže klient nie je schopný detekovať priamo na servere, či daná správa bola už v minulosti prečítaná. Jediná takáto možnosť detekcie je cez príkaz `DELE`, ktorý označí danú správu ako vymazanú a po úspešnom odhlásení zo servera sa táto správa vymaže. Mnohé z klientských aplikácií ošetrujú tento nedostatok rôznymi implementačnými spôsobmi, ktorých úlohou je zapamätanie si správu, ktorá bola už prečítaná, bez potreby túto správu zmazazať na strane servera. Tým pádom predchádzajú znovu uloženiu rovnakej správy na lokálny počítač. V aplikácií implementovanej ako súčasť tejto bakalárskej práce, je tento problém riešený zapamätávaním `Message-ID`, čo je jedinečný identifikátor správy na danom servere. Bližšie sa o implementácií tejto súčasti dozviete v kapitole 6.1. Taktiež nie je možné selektívne prečítať určitú časť správy ako je popísaná v štandarde MIME.

Klientské aplikácie pracujúce s protokolom POP3 musia tiež riešiť ukladanie správ na lokálny počítač a členenie týchto správ do adresárov, tak aby vyhovovali požiadavkám

užívateľa.

V zásade sa protokoly líšia prístupmi. Zatiaľ čo IMAP protokol využíva online prístup pre prácu s jednotlivými adresármi na servere, tak protokol POP3 využíva offline prístup.

Servere pracujúce s protokolom IMAP sú zložitejšie na implementáciu, kvôli množstvu príkazov, na ktoré musia prevádzať a na ktoré musia byť schopné adekvátne odpovedať, a takisto na zložitosť príkazov zasielaných klientskými aplikáciami. Na druhej strane sú IMAP servery náročnejšie na výpočetný výkon, pretože klienti manipulujú s dátami priamo na strane servera.

Výhody a nevýhody protokolov vykresľuje nasledujúca tabuľka:

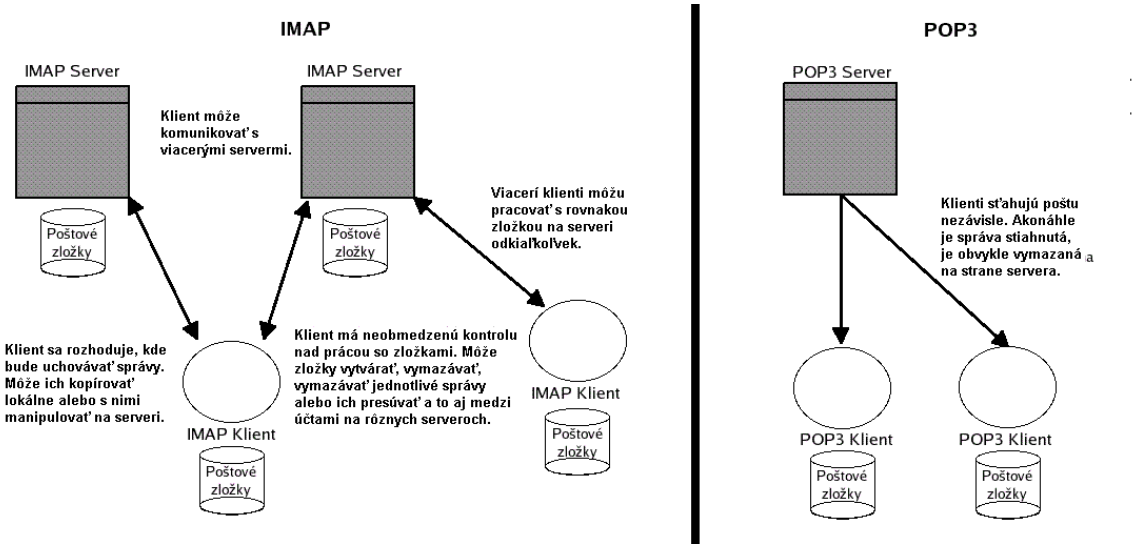
	Výhody	Nevýhody
POP3	Rýchlosť spracovania správ Správy sú uložené na pevnom disku klientského počítača, nezaberajú miesto na servere Veľmi rozšírený a podporovaný poštovými servermi Podporovaný veľkou väčšinou klientských programov	Nie sú dostupné žiadne iné schránky okrem prijatých správ Nejednotnosť formátu ukladaných správ klientmi Neprenositeľnosť uloženej pošty medzi klientmi
IMAP	Správy sú uložené na serveroch, sú prístupné všetkým klientom Prístup ku všetkým zložkám na servere Ľahko implementovateľné spam filtre na serveroch Viacero klientov môže pristupovať ku rovnakým zložkám (prenositeľnosť)	Pomalšia manipulácia so správami na servere Citlivá otázka ukladania dát na servere (zahltenie pamäťového média) Nie je podporovaný veľkou časťou poskytovateľov poštových služieb Nie je podporovaný každým klientom

Informácie boli čerpané z nasledujúcej literatúry [11]. Názorné zobrazenie rozdielu medzi obidvoma protokolmi je na obrázku 3.1.

3.4 Zhrnutie

Cieľom tejto kapitoly bolo štúdium spôsobu získavania elekronickej pošty z POP3 serverov. Boli tu diskutované obmedzenia protokolu, ktoré je nutné riešiť v implemntácií programov pracujúcich nad týmto protokolom.

Takisto tu bol protokol POP3 porovnaný so svojím mladším predchodcom protokolom IMAP a zhrnuté výhody a nevýhody obidvoch protokolov.



Obrázok 3.1: Porovnanie práce prolokolov IMAP a POP3

Kapitola 4

Procesy s premennou periódou

Ďalšou dôležitou súčasťou tejto bakalárskej práce je skúmanie spúšťania procesov s rôznymi časovými rozdielmi. Poznatky nadobudnuté počas štúdia materiálov tejto kapitoly boli použité pri implementácií aplikácie špecifikovanej v zadaní bakalárskej práce.

Táto kapitola diskutuje o procesoch spúšťaných s premennou periodou, ktorá môže byť zabezpečená samotným procesom alebo službami operačného systému. Jednou z takýchto služieb je aj služba cron, ktorá je implementovaná pre operačné systémy Unix/Linux. Ďalším možným prístupom je periodicita spúšťania segmentov kódu, ktorá je kontrolovaná samotným programom.

Na záver budú zhodnotené obidva prístupy a budú diskutované výhody a nevýhody postupu použitého v implementovanej aplikácií.

4.1 Služba cron

Cron je služba, ktorá umožňuje užívateľom spúšťať programy alebo skripty v určitý dátum a čas, alebo v určitých periodických intervaloch. Záznamy o procesoch, ktoré majú byť spúšťané službou, sú uložené v súboroch nazvaných crontabs. Každý užívateľ má možnosť definície vlastného nezávislého súboru crontab. Záznamy v súbore crontab kontroluje proces pod názvom crond. Táto kontrola prebieha štandardne periodicky každú minútu. Proces crond je v stave spánku až do chvíle, kedy nenastane čas definovaný určitým záznamom v crontab súbore.

4.1.1 Nastavenie crontab

Štandardne sa súbory crontab nachádzajú v adresári `/var/spool/cron/crontabs`. Súbory nemusia byť editované priamo, ich editáciu alebo vytvorenie spustí príkaz `crontab -e`. Každý záznam v súbore typu crontab predstavuje jeden príkaz vyvolaný alebo vyvolávaný v definovaný čas. Formát záznamu je rozdelený do šiestich sekcií. Jednotlivé sekcie nesú nasledujúci význam:

1. Minúta: Hodnoty od 0 do 59

2. Hodina: Hodnoty od 0 do 23

3. Deň v mesiaci: Hodnoty od 1 do 31

4. Mesiac: Hodnoty od 1 do 12

5. Deň v týždni: Hodnoty od 0 do 6 (kde 0 predstavuje nedeľu)

6. Príkaz na prevedenie

Medzi jednotlivými sekciami záznamu sú medzery. V rámci jednotlivých sekcií medzery byť nemôžu.

Je takisto možné použiť viacero inštancií jednotlivých sekcií, pomocou čiarky (,) je možné jednotlivé inštancie vymenovávať a pomocou pomlčky je možné určiť sled za sebou idúcich hodnôt v sekcíí. Hviezdička označuje všetky hodnoty v danej sekcii.

Výsledný príkaz može mať aj nasledujúci tvar:

```
5,35 * * * 1-5 rm /home/username/temp/*
```

Príkaz v rámci tohoto záznamu sa spustí každú 5. a 35. minútu, každej hodiny, ľubovoľný deň v mesiaci, ľubovoľný mesiac, od pondelka do piatka. Príkaz vymaže všetky súbory v zložke `temp/` domovského adresára.

Nasledujúce riadky uvádzajú prehľad základných prepínačov programu crontab:

`-e` edituje crontab súbor, v prípade neexistencie tohoto súboru vytvorí nový.

`-l` vypíše všetky záznamy crontab súboru prihláseného užívateľa.

`-r` vymaže crontab súbor prihláseného užívateľa.

`<meno_súboru>` vytvorí crontab súbor zadaného mena pre aktuálne prihláseného užívateľa.

Viac informácií o službe cron a o aplikácií crontab je možné sa dozvedieť v tomto odkaze na literatúru [4].

4.2 Periodicita štartu procesu kontrolovaná samotným procesom

Tento prístup umožňuje samotnému procesu prejsť do režimu spánku na presne definovaný čas. Pri tomto prístupe sa v jazyku C používa funkcia `sleep()` a jej podobné, ktorá je definovaná v knižnici *unistd.h*. Jediným argumentom tejto funkcie je doba v sekundách, na ktorú má byť proces v nečinnosti. Po uplynutí tejto doby proces pokračuje od miesta, kde bola táto funkcia zavolaná. Návratovou hodnotou tejto funkcie je počet sekúnd, ktoré zostávali procesu na opätovné zobudenie. Takže ak prebehne celá doba zadaná argumentom funkcie `sleep()`, funkcia vráti nulu. Knižnica *unistd.h* obsahuje širší sortiment nástrojov pre prácu s uvedením procesu do stavu spánku a zobudením procesu. Do pozornosti by

som rád uviedol aj knižnicu *time.h*, ktorá pracuje s nástrojmi pre získavanie reálneho času. Takisto sa dajú v tejto knižnici nájsť nástroje s prácou napríklad aj s časovačmi (angl. timer).

Bližšie informácie o spomínaných knižniciach a veľa iných informácií o knižniciach použiteľných v jazyku C nájdete tu [5].

4.3 Zhrnutie

V tejto kapitole boli priblížené možnosti riešenia spúštania procesov s premennými periódami. Požitie jedného z týchto prístupov alebo nejakého odlišného je veľmi špecifické a závisí od účelu aplikácie a od skúseností a schopností samotného programátora.

Prvým prístupom je vhodné riešiť problémy všeobecnejšieho charakteru, kde nie sú kladené veľké požiadavky na aplikáciu. Zatiaľčo použitie druhého prístupu je v rukách samotného programátora. Výhodou druhého prístupu je samozrejme aj nezávislosť od nainštalovaných súčastí samotného operačného systému, tzn. väčšia nezávislosť od inštalácie operačného systému.

Pri implementácií aplikácie bola použitá koncepcia druhého prístupu popísaného v tejto kapitole. K tomuto prístupu viedla snaha o užívateľsky zrozumiteľnejší spôsob zadávania údajov aplikácií, ale aj jednoduchosť návrhu aplikácie.

Kapitola 5

Bezpečné ukladanie citlivých informácií

V zadaní tejto bakalárskej práce je aj štúdium manipulácie s citlivými dátami v prostredí operačného systému Linux.

Samotná otázka bezpežnosti je jednou z najdiskutovanejších tém pri modernom vývoji aplikácií, či správy systémov. Nedielnou súčasťou tejto bakalárskej práce je aj táto otázka. V tejto kapitole budú opísané najpoužívanejšie metódy v modernej kryptografií ako aj príklad ich aplikácie v operačnom systéme Linux. Postupne prejdeme rozdelením najdôležitejších typov kryptografických algoritmov, ich históriou ako aj ich použitím v oblastiach počítačovej techniky. Ku každému rozdeleniu budú spomenuté aj najznámejšie mená algoritmov.

Ďalej si v tejto téme spomenieme aj štandard *OpenPGP* pre šifrovanie dát a predstavíme si jednu z aplikácií plne kompatibilnú s týmto štandardom.

V tejto téme je spomenutá aj vlastná jednoduchá implementácia používajúca jednu z nižšie uvedených metód. Tento algoritmus je súčasťou implementácie zadanej aplikácie.

5.1 Symetrické šifrovanie

Princíp tejto metódy spočíva v existencií jedného tajného kľúča, pomocou ktorého bola informácia zašifrovaná a takisto ju je možné zo šifry pomocou tohoto kľúča dešifrovať. Z tohto vyplýva nutnosť pred začiatkom komunikácie odoslať dôveryhodným kanálom šifrovací kľúč spolu s ďalšími údajmi (napr. konkrétny typ algoritmu) druhej strane. Problémom tohto riešenia je, že nie je možné zaistiť tzv. nepopierateľnosť zodpovednosti (nie je možné jednoznačne určiť autora správy, lebo obaja komunikujúci partneri majú totožný šifrovací kľúč). Kľúč v tomto šifrovaní sa často distribuje pomocou použitia asymetrického šifrovania.

Výhodou tohoto spôsobu šifrovania dát je nízka náročnosť algoritmov na výpočtový výkon počítača. Preto ich použitie prevláda hlavne v spracovaní veľkých tokov dát.

Symetrická kryptografia sa ďalej delí na prúdové šifry, ktoré informáciu spracovávajú

po bitoch a blokové šifry, ktoré informáciu rozdelia na rovnaké bloky (obvykle 64 bitové) a tie spracovávajú na výslednú šifru.

Princíp práce algoritmov pre symetrické šifrovanie približuje obrázok 5.1.

Informácie o symetrickom šifrovaní boli čerpané hlavne z nasledujúcich stránok [13] a odkazov na týchto stránkach.

5.1.1 Blokové šifrovanie

Blokové šifra pracuje s pevne stanoveným počtom bitov, tzv. blokov. Podstatné je, že všetky bloky informácie sú šifrované tou istou transformáciou a dešifrované taktiež tou istou transformáciou.

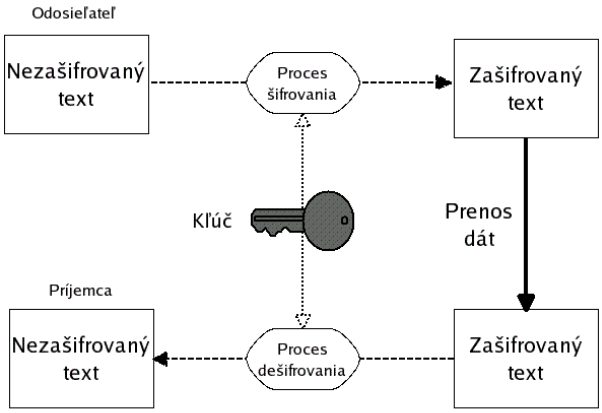
Medzi najznámejšie algoritmy, pracujúce na princípe blokového šifrovania patrí DES a Blowfish.

5.1.2 Prúdové šifrovanie

Tento spôsob šifrovania spočíva v zmene časti informácie. Každá táto časť sa príslušnou transformáciou zmení za jednotku času. Výhodou tohto šifrovanie je fakt, že ak počas prenosu zašifrovanej informácie dôjde k chybe. Zmenia sa len znaky, počas ktorých táto zmena nastala.

Tieto šifry sa používajú hlavne na šifrovanie veľkých objemov dát nepoznanej veľkosti napr.: v bezdrôtových pripojeniach.

Medzi najznámejšie algoritmy využívajúce prúdové šifrovanie sú RC4, A5/1 alebo A5/2.



Obrázok 5.1: Ukážka princípu práce symetrického šifrovania

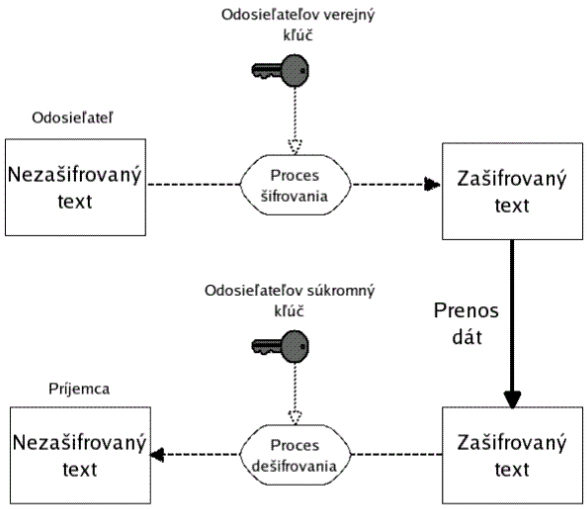
5.2 Asymetrické šifrovanie

V tejto kryptografickej metóde sa na šiforvanie a dešifrovanie informácie používajú rozdielne kľúče. Kľúč, ktorým je správa šifrovaná sa nazýva verejný kľúč. Príjemca správy nemusí tento kľúč vôbec poznať. Naopak kľúč, ktorým je správa dešifrovaná sa nazýva súkromný kľúč, ktorý odosielteľ správ zpravidla nepozná. Tento druh kryptografie sa tiež používa aj pre elektornický podpis.

Tento prístup šifrovania dát je náročnejší na výpočetné zdroje ako symetrické šifrovanie. Medzi dôležité vlastnosti tohoto šifrovania patrí aj nemožnosť odvodenia jedného z kľúčov zo znalosti kľúča druhého. Tento spôsob zabezpečuje autenticitu dát, tj. ak správu môžeme dešifrovať niekoho verejným kľúčom, máme záruku, že to šifrovala práve táto osoba. Viacej informácií nájdete tu [3] a v príslušných odkazoch na tejto stránke.

Najznámejšími algoritmami pracujúcimi na báze tohto spôsobu šifrovania dát sú RSA, ElGamal a Diffie-Hellman.

Spôsob šifrovania ponocou asymetrických šifier je na obrázku 5.2.



Obrázok 5.2: Ukážka princípu práce asymetrického šifrovania

5.3 Hashovacie funkcie

Sú formou kryptografických funkcií, ktoré ľubovoľne dlhý reťazec prevedú na reťazec pevnej dĺžky tiež nazývaný odtlačok prsta. Na hashovacie funkcie sú kladene nasledujúce požiadavky:

- Neexistuje fumkcia, ktorá by previedla odtlačok prsta späť na vstupný reťazec.
- Neexistujú dva vstupné reťazce, ktoré by mali rovnaký odtlačok prsta.

- Ak sa zmení jeden bit vo vstupnom reťazci, výsledný odtlačok prsta sa musí zmeniť viac ako v jednom bite oproti pôvodnému vstupnému reťazcu.

Vzhľadom k tomu, že tieto algoritmy sú jednocestné, slúžia na overenie autentizácie užívateľov.

V tejto kategórií sú najznámejšími funkciami MD5 a SHA-1. Tento spôsob šifrovania sa používa hlavne na kontrolu správnej autentifikácie.

5.4 GNU Privacy Guard

Táto časť sa venuje možnosti použitia softvéru, ktorý slúži hlavne na šifrovanie a dešifrovanie súborov, správ a digitálny podpis.

Tento program a jemu podobné vychádzajú z programu *PGP*, ktorého publikovanie bolo významným miľníkom vo svete šifrovania.

Program *PGP*, slúžiaci na šifrovanie postavené na základoch asymetrickej kryptografie, bol vydaný už v roku 1991. Mal obrovský vplyv vo svete kryptografie, preto bol neskôr prijatý ako otvorený štandard. Nižšie sa budem venovať programu, ktorý je plne kompatibilný s otvoreným *OpenPGP* štandardom.

5.4.1 Charkateristika GPG

GPG predstavuje softvérovú náhradu za *PGP*, je vydaný pod licenciou *GPL*. Je taktiež súčasťou *Free Software Foundation*[1]. Táto implementácia je úplne kompatibilná so štandardom *OpenPGP*.

GPG je hybridný šifrovací program, ktorý používa kombináciu konvencií symetrickej (rýchlosť) a asymetrickej kryptografie (ľahkosť výmeny kľúčov).

Tento program je stabilný, kvalitný softvér používaný vo voľne šíriteľných operačných systémoch, taktiež je využitý na poštových systémoch ako Horde a v prehliadačoch Mozilla Thunderbird.

5.4.2 História

GPG bol pôvodne vyvinutý Wernerom Kochom v roku 1999 a bol vydaný vo verzií 1.0. Program podporovala nemecká vláda a posunula jeho implementáciu do Microsoft Windows v roku 2000.

Druhá verzia programu bola vydaná v roku 2006, kde boli značné zmeny v architektúre implementácie oproti predošlým verziám.

[1] Nezisková organizácia na podporu hnutia pre slobodný softvér.

5.4.3 Princíp práce GPG

Program šifruje dáta pomocou využitia asymetrických kľúčov generovaných GPG užívateľmi. Výsledné kľúče môžu byť vymieňané medzi užívateľmi rôznymi spôsobmi. Vždy však musí byť zaistená bezpečná výmena kľúčov kvôli útokom typu *spoofing*[2].

Aplikácia umožňuje takisto digitálny podpis správ, čo umožňuje verifikáciu odosieľateľa správy a jej samotnú integritu, v prípade, že spáva bola spoľahlivo doručená.

Samotné *GPG* nevyužíva patentovaný alebo ináč chránený softvér alebo algoritmy. Nepoužíva ani šifrovací algoritmus *IDEA*[3], ktorý bol využívaný v *PGP* od samotného začiatku. Namiesto neho používa celú radu nepatentovaných algoritmov.

5.5 Použitie vo vlastnej implementácií

Vo vlastnej aplikácií bolo použité jednoduché šifrovanie postavené na znalosti symetrického šifrovania a prúdových šifier.

Aplikácia vygeneruje náhodný kľúč, ktorý je prístupný algoritmom na šifrvanie a dešifrovanie. Tento kľúč má dĺžku 128 bitov. Po zadaní hesla je každý bajt hesla vystavený príslušnému bajtu kľúča pomocou operácie XOR (Exclusive OR). Ku každému bajtu takto vzniknutého reťazca je pripočítaná konštanta.

Pri dešifrovaní je táto konštanta odpočítaná a výsledný reťazec je vystavený tomu istému kľúču bajt po bajte operáciou XOR.

5.6 Zhrnutie

V tejto kapitole boli študované možnosti bezpečného ukladania citlivých informácií. Taktiež bol predstavený široko používaný program *GPG*, slúžiaci na rôzne účely šifrovania. Na záver kapitoly boli znalosti získané štúdiom tejto časti zadania bakalárskej práce predvedené na jednoduchom algoritme implementovanom v zadanej aplikácií.

Informácie v tejto kapitole boli čerpané aj z [15].

[2] Podhodenie falošných údajov

[3] Šifrovací algoritmus pracujúci na princípe blokovej šifry

Kapitola 6

Popis implementácie aplikácie IMAP proxy

V tejto kapitole sa budem venovať samotnej implementácií aplikácie pre sťahovanie elektronickej pošty z rôznych POP3 serverov a jej následné ukladanie a členenie do adresárov na užívateľom definovaný IMAP server. Celá manipulácia s poštou z určitého POP3 servera prebieha v uživateľom definovanom čase.

Postupne sa budeme zaoberať princípom činnosti aplikácie, ako aj činnosťou jednotlivých modulov, konfiguráciou aplikácie a popíšeme si aj súbory vytvaráné počas činnosti aplikácie.

Takisto tu bude spomenuté aj prepojenie s implementáciou pomocnej aplikácie, ktorá demonštruje alternatívu pre bezpečné uloženie hesiel pre prístup k POP3 serverom.

6.1 Návrh aplikácie

Aplikácia je naimplementovaná podľa zvyklostí jazyka C. Pri preklade je použitý štandard `gnu99`. V aplikácií je použité paradigma modulárneho programovania, kde je vyzdvihnutá snaha o čo najlepšiu dekompoziciu danej problematiky. Jednotlivé moduly pracujú nad určitým typom problému. Rozhranie každého modulu programu je obsiahnuté v hlavičkovom súbore, pomocou týchto rozhraní jednotlivé moduly programu medzi sebou komunikujú.

Pri implementácií sú s výhodou použité funkcie určené pre komunikáciu so servermi, ktoré nezaťažujú limit vyrovnávacej pamäte určený pre soket v jadre operačného systému. Tieto funkcie slúžia pre posielanie dát na server a následne aj na ich prijímanie. Implementácie týchto a podobných funkcií sú opísané v knihe [16]. Použité funkcie sa volajú `readn` (pre prijímanie dát zo servera) a `writen` (pre posielanie dát na server).

Ihneď po spustení je aplikácia uvedená do práce na pozadí. Špecifikácia serverov, z ktorých má prebiehať sťahovanie pošty, ako aj špecifikácia servera, pre ktorý má byť pošta organizovaná, sú zaistené v konfiguračnom súbore s názvom `imapproxy.conf`.

Prečítanie konfigurácie a jej následné uloženie do pamäte prebieha vždy pri štarte aplikácie. Je opakované vždy, keď sú spracované všetky položky v predchádzajúcom čítaní konfiguračného súboru.

Následne na to sú utriedené jednotlivé položky podľa časov spustenia manipulácie s poštou a uložené do štruktúr. V každej štruktúre sú uložené nasledujúce údaje, ktoré su aj kľúčovými slovami v konfiguračnom súbore:

MAILBOX: Názov poštového adresára, do ktorého sa má pošta ukladať na IMAP server.

SERVER: Adresa POP3 servera, z ktorého je získavaná elektronická pošta.

PORT: Číslo portu služby POP3 servera, pracujucého na zadanej adrese.

USER: Užívateľ, ktorého pošta sa má stiahnúť z POP3 servera.

DELETE: Príznak, či sa má stiahnutá pošta z POP3 servera vymazať.

FLAGS: Príznaky, pod ktorými má byť stiahnutá pošta uložená na IMAP server.

TIME: Čas, kedy má byť s poštou manipulované. Týchto položiek môže byť v súbore s konfiguráciou ľubovoľné množstvo.

Takisto sa z konfiguračného súboru prečítajú a uložia do pamäte údaje o IMAP servere, na ktorom má byť pošta organizovaná. Tieto informácie sú nasledujúce:

IMAP_SERVER: Adresa IMAP servera.

IMAP_PORT: Číslo portu pre IMAP server.

IMAP_USER: Meno užívateľa IMAP servera.

IMAP_PASSWORD: Heslo pre zadaného užívateľa.

Samotné načítavanie prebieha len do pomocnej štruktúry. Tá je súčasťou štruktúry, ktorá je doplnená o počet sekúnd od polnoci 1. januára 1970[1].

Tu nastávaju prípady, kedy je rovnaká konfigurácia uložená vo viacerých štruktúrach, tieto sa odlišujú len spomínanou časovou informáciou. Akonáhle nastane čas vykonávania určitej položky, aplikácia sa pripojí na daný POP3 server a začne sa sťahovať každá správa zvlášť a následne sa uloží do definovanej zložky na IMAP servere.

Podľa nastavenia položky sa správa vymaže alebo ostáva na POP3 servere aj po manipulácií s ňou. Po jej stiahnutí a načítaní do pamäte sa zo správy získava jej jedinečný identifikátor, ktorým je `Message-ID`. Podľa tohto jedinečného označenia zadanej správy prebieha kontrola v súboroch, ktoré sa vytvárajú alebo editujú zároveň so spracovaním príslušnej položky. Tieto súbory sú pomenované podľa mena užívateľa a zložky, kde sa má

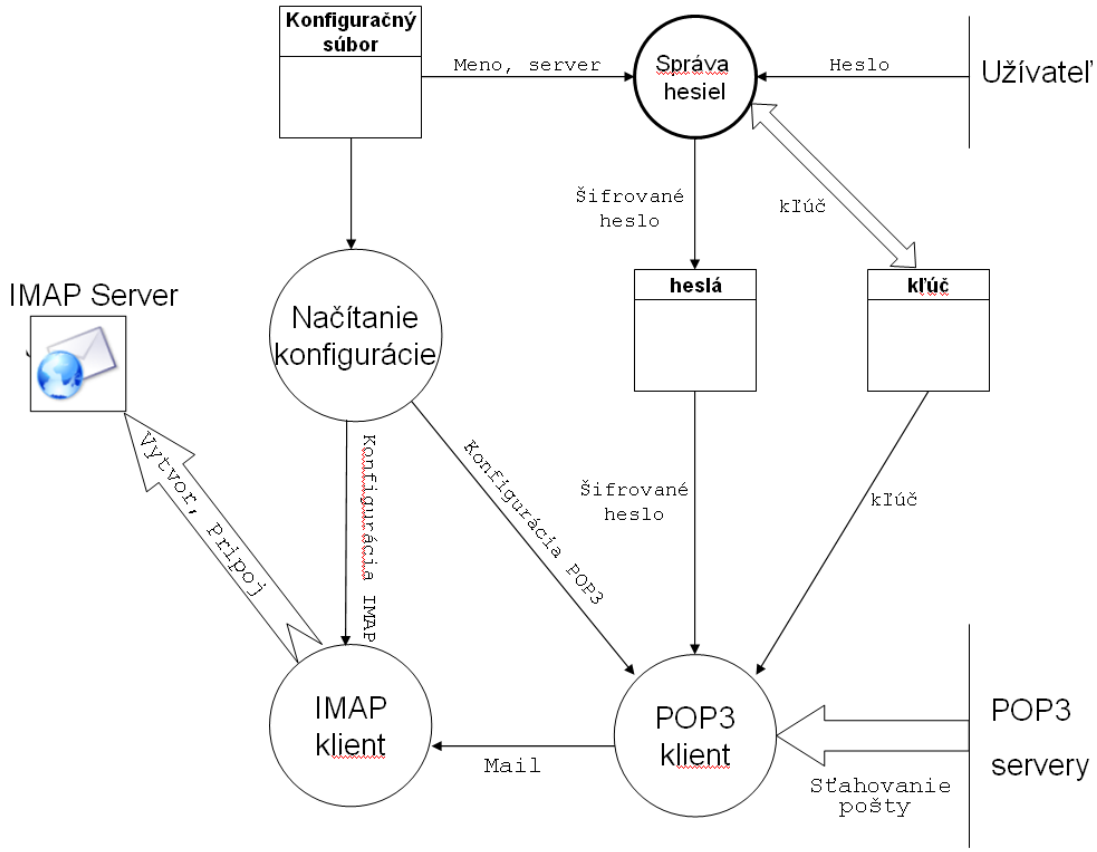
[1] Čas, kedy začala tzv. Unixová epocha

pošta ukladať. Názvy týchto súborov sú tvorené následovne:
`<názov_zložky>.<meno_užávateľa>`.

Záznamy `Message-ID` sú v nich uložené po riadkoch.

Pred pripájaním sa na POP3 server sa v súbore `imap.passwd` skontroluje prítomnosť príslušného hesla pre zadané užívateľské meno na tomto servere. Heslo je v súbore uložené podľa názvu servera a užívateľa. Následne prebieha odšifrovanie tohto hesla spôsobom popísaným v stati 5.5 a jeho poslanie na POP3 server.

Činnosť aplikácie a jej celkový návrh je na obrázku 6.1. Tu je zobrazená aj časť *Správa hesiel*, ktorá predstavuje samostatnú aplikáciu komunikujúcu s užívateľom, ktorá umožňuje pridávať heslá do súboru. Táto aplikácia bude diskutovaná v kapitole 7.



Obrázok 6.1: Štruktúra aplikácie a spôsob komunikácie medzi jednotlivými modulmi

Nasleduje popis implementácie a popis činnosti jednotlivých modulov aplikácie.

6.2 Popis implementácie načítania konfiguračného súboru

Modul pre načítanie hodnôt z konfiguračného súboru a pre ich ukladanie do štruktúr má názov `parser.c` a komunikuje s ostatnými modulmi cez svoje rozhranie nazvané `parser.h`.

Tento modul slúži na uloženie iformácií zadaných užívateľom do špecifických dátových štruktúr. Algoritmus prechádza konfiguračným súborom po riadkoch a hľadá kľúčové slová. Ak nájde kľúčové slovo priradí k nemu hodnotu oddelenú ľubovoľným počtom prázdnych znakov. Takisto sa v tomto module nastavujú implicitné hodnoty pre nepovinné položky.

Po správnom uložení všetkých povinných nastavení pre účet na POP3 servere sa tieto nastavenia uložia do poľa a usporiadajú sa podľa veľkosti časovej hodnoty spustenia od najmenšej položky po najväčšiu. Na tento účel bola použitá funkcia `qsort`, ktorú je možné nájsť v štandardnej knižnici jazyka C.

Ako bolo už povedané vyššie konfigurácia je uložená v súbore s názvom `imapproxy.conf`. Príklad konfigurácie súboru vyzerá takto:

```
#configuration to IMAP server
IMAP_SERVER: localhost
IMAP_PORT: 143  #default 143
IMAP_USER: miro
IMAP_PASSWORD: password

#configuration to POP3 servers
MAILBOX: seznam
  SERVER: pop3.seznam.cz
  PORT: 110  #default 110
  USER: user1
  DELETE:   #Y/N, default N
  FLAGS: DrAS
#\Seen: S
#\Answered: A
#\Deleted: De
#\Flagged: F
#\Draft: Dr
  TIME: Wed 17.30  #format ‘‘Day HH.MM’’
  TIME: Mon 20.00
  TIME: Sun 06.00
END_MAILBOX

MAILBOX: zoznam
  SERVER: pop.zoznam.sk
  PORT:
  USER: user2@zoznam.sk
  DELETE: Y
  FLAGS:
```

```
  TIME: Mon 12.00
  TIME: Tue 08.00
  TIME: Thu 16.00
END_MAILBOX
```

Konfiguračný súbor pozostáva z dvoch hlavných častí. Tie budú popísané v nasledujúcich častiach. Začiatok komentárov je v tomto súbore označovaný znakom `#` a ich platnosť je do konca daného riadka.

Predchadzajúci príklad znamená, že správy sa majú sťahovať z dvoch POP3 serverov a ukladajú sa na lokálny IMAP server. Tu sa ukladajú do zložiek `seznam`, pre prvý z nich a `zoznam` pre druhý.

Tu podám stručné vysvetlenie významu jednotlivých prvkov prvého záznamu. Prvým je adresa servera, z ktorého bude získavaná pošta pomocou protokolu POP3. V tomto prípade `pop3.seznam.cz`. Služba POP3 servera beží na vzdialenom počítači na porte `110` a prihlasovaice meno užívateľa je `user1`. Ďalej stiahnuté správy z tohto účtu nebudú vymazané (implicitné nastavenie). Na server IMAP sa správy uložia s príznakmi rozpísaná, odpovedaná a videná. Postup sťahovania pošty prebieha každú stredu o 17.30, každý pondelok o 20:00 a každú nedeľu o 6:00.

6.2.1 Zadanie konfigurácie IMAP servera

Toto nastavenie sa môže v konfiguračnom súbore objaviť aj viackrát. No nastavenie pre server IMAP sa uloží do predom definovanej štruktúry vždy len raz a to pri zistení prvého výskytu jednotlivých kľúčových slov. Jediným údajom, ktorý je predom definovaný a môže byť vynechaný je číslo portu, ktorého prednastavená hodnota je *143*. Jednotlivé kľúčové slová tohto nastavenie sa môžu nachádzať kdekoľvek v súbore.

6.2.2 Zadanie konfigurácie POP3 účtov

Nastavenia jednotlivých poštových adresárov pre IMAP server sú oddelené medzi kľúčovými slovami `MAILBOX:` a `END_MAILBOX:`. Pri druhom zo spomínaných je kontrolovaná prítomnosť zadania všetkých povinných častí nastavenia pre jeden POP3 účet. Medzi týmito kľúčovými slovami sa môže nachádzať aj viac jednotlivých údajov, avšak implementovaný algoritmus vždy berie v úvahu len prvý výskyt danej hodnoty kľúčového slova. Výnimkou je údaj o čase spustenia manipulácie s poštou pre danú zložku. Tu sa akceptujú všetky výskyty tohto údaju a ukladajú sa do danej štruktúry.

Čo sa týka počtu nastavení jednotlivých POP3 účtov je ich počet obmedzený len na systémové prostriedky počítača.

Medzi povinné položky patrí meno servera, názov poštovej zložky, prihlasovacie meno užívateľa a minimálne jeden výskyt hodnoty v položke `TIME:`. Samozrejme podmienkou pre úspešné zadanie je aj zadanie kľúčového slova `END_MAILBOX:` vždy na konci definície POP3 účtu.

Nepovinné položky účtov majú nastavené svoje implicitné hodnoty. Pre číslo portu je to hodnota *110*. Ak je vynechaný údaj o príznakoch správy posielaných IMAP serveru, klient vkladá do definovaného adresára správu bez špecifikácie tejto položky. Ďalej závisí len od implemntácie servera, ako tieto príznaky pre danú správu nastaví. Poslednou nepovinnou položkou je údaj o odstránení stiahnutej správy z POP3 účtu. Prednastavená je hodnota *N*, čo znamená, že správy nebudú vymazané.

Príznaky správ ukladaných na IMAP server majú v konfiguračnom súbore nasledujúci tvar:

Seen: Správa je označená ako prečítaná (skr.: `S`).

Answered: Správa je označená ako odpovedaná (skr.: `A`).

Draft: Správa je označená ako rozpísaná (skr.: `Dr`).

Deleted: Správa je označená ako vymazaná (skr.: `De`).

Flagged: Správa je označená v zložke na servere (skr.: `F`).

Skratky týchto príznakov sa zapisujú ako jeden reťazec do konfiguračného súboru v ľubovoľnej kombinácií. Príklad:
`FLAGS: SADe`
Správy s týmto príznakom budú označené ako prečítané, už odpovedané a vymazané.

Formát času je zapisovaný v tvare `TIME: <deň >hh.mm`, pričom deň je deň v týždni a vyjadruje sa prvými tromi písmenami jeho anglického prekladu. Prvé písmeno je veľké. Čas sa vyjadruje v hodinách a minútach, tieto sú oddelené bodkou (.). Medzi dňom a konkrétnym časom môže byť ľubovoľný počet medzier.

Tento modul predstavuje základ komunikácie aplikácie s klientom.

6.3 Popis implementácie komunikácie pomocou POP3 protokolu

V tejto kapitole sa budem venovať analýze spôsobu zasielania príkazov pre POP3 server, rozdeleniu typov odpovedí POP3 servera a tiež tu budú spomenuté príkazy ktoré boli v tejto komunikácií použité. Taktiež je v tejto časti implementovaná aj funkcia, ktorá z danej správy získa hodnotu jej jedinečného identifikátora `Message-ID`. Táto komunikácia je implementovaná v module s názvom `pop3.c` a pre prístup k jej funkciám a dátovým typom sa používa rozhranie `pop3.h`.

6.3.1 Analýza zasielania príkazov pre POP3 server

Každý príkaz pre POP3 server sa skladá z troch alebo štyroch znakov a z prípadných parametrov jednotlivých príkazov. Na server sa posiela vždy celý riadok príkazu vrátane

znakov `CRLF` na jeho konci. Jednotlivé príkazy možu byť dlhé najviac 40 znakov. Pri zadávaní príkazov nie je dôležité, či sú poslané serveru veľkými alebo malými písmenami, príkazy protokolu nie sú tzv. `case-sensitive`.

6.3.2 Odpovede POP3 serverov

Väčšina odpovedí POP3 serverov obsahuje len jeden riadok. Server odpovedá dvoma spôsobmi. Prvý je v prípade pozitívnej odpovede, kedy na začiatku odpovede na príkaz je sekvencia znakov `‘+OK´`. Pri negatívnej odpovedi je na začiatku sekvencia znakov `‘-ERR´`. Vzhľadom na jednoznačnosť protokolu, kontrolujem v aplikácií výskyt len kladnej odpovede.

V prípade viac riadkových odpovedí, je odpoveď ukončená znakmi `CRLF.CRLF`. Sada znakov `.CRLF` sa nepokladá za časť viacriadkovej odpovede. Príklad príkazu kedy sa očakáva v prípade úspešného prevedenie viacriadková odpoveď je príkaz `RETR`, ktorý slúži na získanie správy. Parametrom tohto príkazu je číslo správy.

Viac informácií o protokole nájdete tu [2].

6.3.3 Kroky použité pri získavaní pošty

Úvodom budú tieto kroky vymenované a následne sa budeme zaobrať popisom jednotlivých krokov. V implementácií prebieha komunikácia s IMAP serverom počas sťahovania pošty, pretože manipulácia prebieha nad jednotlivými správami.

1. Nadviazanie komunikácie s POP3 serverom a prihlásenie užívateľa.
2. Zistenie počtu správ na servere.
3. Odhlásenie užívateľa a ukončenie komunikácie s POP3 serverom.
4. Prihlásenie na IMAP server.
5. Vytvorenie príslušnej zložky na IMAP servere.
6. Začiatok cyklu pre sťahovanie jednotlivých správ.
   (a) Nadviazanie komunikácie s POP3 serverom a prihlásenie užívateľa.
   (b) Zistenie veľkosti určitej správy.
   (c) Stiahnutie správy a načítanie `Message-ID` z jej obsahu a uloženie správy aj `Message-ID` do spoločnej štruktúry .
   (d) Prípadné vymazanie správy (závisí na nastavení v konfiguračnom súbore).
   (e) Uloženie správy do príslušnej zložky na IMAP server.
   (f) Odhlásenie užívateľa a ukončenie komunikácie s POP3 serverom.
7. Koniec cyklu.

8. Odhlásenie užívateľa z IMAP servera a ukončenie komunikácie s ním.

Časté prihlásenie a odhlásenie na server prebieha z časových dôvodov. V testoch sa zistilo, že trvanie uloženia správ na IMAP server je v niektorých prípadoch dostatočné na prekročenie doby nečinnosti potrebnej na automatické odhlásenie z POP3 servera. Takto sa vyhýbam nepríjemnosťam spojenými s touto skutočnosťou. Automatické odhlásenie implementované v IMAP serveroch má štandardne nastavenú dlhšiu časovú kvôtu nečinnosti.

Ďalej budem popisovať jednotlivé kroky v algoritme sťahovania správ:

Po nadviazaní spojenia so serverom nám server pošle informáciu, že je ochotný komunikovať, v opačnom prípade to prevdepodobne nie je adresa POP3 servera alebo je v konfiguračnom súbore zadané nesprávne číslo portu. Prihlásenie užívateľa sa zadáva príkazom `USER`, ktorého parametrom je prihlasovacie meno užívateľa. Po správnom prihlásení je potrebné zadať heslo príkazom `PASS`. Heslo je získané v šifrovanej podobe zo súboru `imap.passwd` a je nutné ho dešifrovať. Súčasťou modulu je aj dešifrovanie tohto hesla. V záujme bezpečnosti je heslo uchovávané v pamäti len na krátku dobu. Popis procesu dešifrovania hesla je popísaný vyššie 5.5.

Následne po prvom prihlásení sa zistí pomocou príkazu `LIST`, ktorý sa posiela bez parametrov, počet správ prítomných v schránke na manipulovanom účte. Po tomto zistení prebieha cyklus podľa počtu správ, ktoré sa majú stiahnuť. Predtým sa však aplikácia prihlási na IMAP server a vytvorí príslušný adresár platný pre daný účet zadaný v konfiguračnom súbore, ak už tento nebol vytvorený v minulosti.

V spomínanom cykle sa aplikácia znovu prihlási na daný POP3 server a príkazom `STAT` zistí veľkosť správy, s ktorou sa následne chystá manipulovať. Parametrom uvedeného príkazu je poradové číslo správy v schránke POP3 servera.

Po zistení veľkosti správy sa serveru pošle príkaz `RETR`. Súčasťou odpovede servera na tento príkaz je aj samotná správa. Jediným parametrom tohto príkazu je číslo správy. Po načítani správy do pamäte, nasleduje získanie `Message-ID` z uloženej správy.

Stiahnutú správu je podľa nastavenia v konfiguračnom súbore možné v tejto časti vymazať príkazom `DELE`, ktorého parametrom je číslo správy. Táto správa sa na servere označí ako vymazaná. Samotné vymazanie správy na POP3 servere prebieha až po úspešnom odhlásení užívateľa.

Potom prebieha pokus o uloženie správy do schránky s určeným názvom na IMAP server a uvoľnenie dát z štruktúry určenej na uloženie správy a odhlásenie sa z POP3 servera. Tento cyklus sa opakuje podľa počtu zistených správ v schránke POP3 servera.

Po ukončení cyklu nasleduje odhlásenie z IMAP servera.

Popis príkazov pre IMAP server a jeho odpovedí je diskutovaný v ďalšej stati.

6.4 Popis implementácie komunikácie s IMAP serverom

V tejto sekcií sa zaoberám analýzou príkazov zasielaných IMAP serveru, ako aj definíciou ich tvaru popísaného v RFC 3501. Takisto budú diskutované typy odpovedí IMAP servera, ako

aj reakcie aplikácie na tieto odpovede. Časti komunikácie s IMAP serverom budú rozdelené do jednotlivých sekcií a podrobne popísané. Funkcie pre prácu s IMAP serrverom sú uložené v module `imap.c` rozhranie pre použitie týchto funkcií je v súbore `imap.h`.

6.4.1 Analýza zasielania príkazov pre IMAP server

Podobne ako v POP3 protokole, každý príkaz pre server IMAP je zasielaný ako riadok ukončený sekvenciou znakov `CRLF`. Narozdiel od predchádzajúceho protokolu je príkazová sada pre komunikáciu so serverom bohatšia, ako aj parametre jednotlivých príkazov majú širšie možnosti, čo prichádza s väčšími možnosťami manipulácie so správami na IMAP serveroch. Každý príkaz pre server začína jedinečným tagom pre tento príkaz v danom sedení (napr.: `A001`). Tag môže byť ľubovoľný alfanumerický reťazec.

6.4.2 Odpovede IMAP serverov

Podľa príkazu poslanému IMAP serveru môže tento odpovedať jedno alebo viacriadkovými odpoveďami. Tu však nájdeme celú radu príkazov, kedy servery odpovedajú mnohoriadkovými odpoveďami. Avšak pri komunikácií tejto aplikácie takéto príkazy nie sú využívané. Narozdiel od protokolu POP3, servery pracujúce pomocou protokolu IMAP odpovedajú tromi základnými spôsobmi. Prvým druhom odpovedi je úspešné vykonanie príkazu, ktoré sa indikuje sekvenciou znakou `‘OK´`. Druhým je nemožnosť prevedenia daného príkazu (napr.: príkaz je platný pre iný stav komunikácie). V tomto prípade server pošle odpoveď obsahujúcu sekvenciu znakov `‘NO´`. Posledný druh je odpoveďou na chybu protokolu (napr.: nepoznaný príkaz alebo zlé parametre príkazu). Každá z týchto odpovedí začína tagom príslušného príkazu. Ďalšou formou odpevede je odpoveď na špecifický príkaz. V tejto aplikácií je to odpoveď na príkaz `APPEND`, kedy server pripravený na prijatie spávy reaguje odpoveďou začínajúcou znakom `‘+´`.

Bližšie informácie o protokole hľadajte tu [1].

6.4.3 Časti komunikácie s IMAP serverom

V tejto časti predstavím príkazy používané mojou aplikáciou ako aj spôsob tvorby parametrov jednotlivých príkazov. Takisto tu budú spomenuté pomocné funkcie, dôležité pri komunikácií s IMAP serverom.

Prihlasovanie na server

Po pripojení, server pošle klientovi uvítaciu správu. Ak správa nebolo poslaná alebo server poslal správu so zápornou odpoveďou, aplikácia sa ukončí. Jedným z dôvodov môže byť nesprávna adresa servera alebo vnútorná chyba na servere. Po skontrolováni tejto správy nasleduje prihlásenie užívateľa na tento server pomocou príkazu `LOGIN`. Parametrami tohto príkazu sú meno a heslo užívateľa. Po príhlásení prechádza sedenie so serverom do autentifikovaného stavu.

Vytváranie adresára na servere

Ďalším príkazom, ktorý aplikácia využíva, je príkaz pre vytváranie poštovej zložky na servere. Aplikácia sa pokúša vytvoriť adresár na danom servere. V prípade neúspechu je s najväčšou pravdepodobnosťou tento adresár už vytvorený. Príkazom pre vytvorenie zložky je príkaz `CREATE`, ktorého jediným parametrom je meno zložky.

Uloženie správy do zložky na server

Tu je použitý príkaz `APPEND`. Využívanými parametrami tohto príkazu sú meno zložky, do ktorej sa má správa uložiť, príznaky správy a veľkosť správy, pričom parameter určujúci príznaky ukladanej správy je nepovinný. Tu je volaná funkcia, ktorá reťazec zadaný ako príznaky správy v konfiguračnom súbore prevedie na text potrebný ako parameter príznak príkazu `APPEND` na server. Podľa návratovej hodnoty tejto funkcie sa tiež rozhodne či vôbec bude nejaký reťazec poslaný.

Ďalším nepovinným parametrom, ktorý však nie je v aplikácií vôbec využitý je voliteľný parameter označujúci časovú značku pre správu. V prípade tejto aplikácie, kedy je tento parameter vynechaný je serverom použitý aktuálny čas. Správnosť prevedenia tohoto príkazu je serverom indikovaná v odpovedi obsahujúcej na začiatku znak `'+'`. Za touto odpoveďou server očakáva od klienta poslanie celej správy.

V tejto časti aplikácia tiež kontroluje pomocou `Message-ID` správy, či už bolo so správou manipulované. Táto kontrola prebieha v súboroch, ktorých popis je diskutovaný vyššie 6.1. V tejto časti aplikácie prebieha aj vytvorenie jednotlivých súborov, prehľadávanie ich obsahu a porovnávanie `Message-ID` aktuálnej správy s obsahom súboru.

Pri ukladaní správy na server tiež prebieha rozpoznávanie reťazca s príznakmi ukladanej správy. Podľa toho, či boli nejaké príznaky rozpoznané sa aplikácia rozhodne, či tento parameter použije v samotnom príkaze.

Odhlásenie zo servera

Odhlásenie zo servera je implementované pomocou príkazu `LOGOUT`, ktorý nemá žiadne parametre. Po úspešnom odhlásení nasleduje uzatvorenie soketu pre komunikáciu s IMAP serverom.

6.5 Možnosti vylepšenia aplikácie

Aplikácia by sa dala vylepšiť rôznymi rozšíreniami, ako je sofistikovanejšie nastavenie časových položiek v konfiguračnom súbore ako napríklad spustenie manipulácie s poštovou v konštantných časových intervaloch. Takisto je možná implemntácia grafického užívateľského rozhrania. Ďalším možným rozšírením môže byť imaplementácia podpory bezpečnej komunikácie pomocou *SSL*. Taktiež zaznamenávanie udalostí v aplikácií by mohlo byť

implementované sofistikovanejším spôsobom. Bezpečnostným nedostatkom aplikácie je ukladanie hesla pre IMAP server ako nezašifrovaného textu do súboru s konfiguráciou.

6.6 Zhrnutie

Meno aplikácie po preklade je `imapproxy` a po jej spustení aplikácia prechádca do práce na pozadí. Udalosti zaznamenané v priebehu chodu aplikácie sú ukladané do súboru `imapproxy.log`.

Aplikácia bola prekladaná a testovaná na operačnom systéme *Fedora Core 5*a na študentskom servere *Merlin*[2], slúžiacom pre potreby študentov FIT. Aplikácia bola preložená prekladačom *GNU Compiler Collection (gcc)* s parametrami `-std=gnu99 -Wall -pedantic -W`. Použitá verzia prekladača je `3.4.6`. Ladenie bolo prevedené pomocou programu *GNU Debugger (gdb)*.

Pri testovaní bol použitý IMAP server bežiaci na lokálnom operačnom systéme, na bezplatnom e-mailovom servere *Atlas*[3], či študentskom serveri *Eva*[4].

Pri testovaní prístupu POP3 boli použité bezplatné e-mailové servery portálov *Seznam*, *Zoznam*, *Azet*, *Pobox* a *Atlas*.

Vo vývoji boli využité znalosti z predmetov *IPK* a *IAS* a to hlavne pri implementácií sieťovej komunikácie pomocou *BSD Sockets*. Viac o implementácií sieťovej komunikácie sa dozviete ve tejto literatúre [16].

Aplikácia spĺňa požiadavky popísané v zadaní tejto bakalárskej práce. V implementácií boli využité poznatky zo štúdia teoretickej časti bakalárskej práce. Samotný návrh bol vytvorený po dohode s vedúcim bakalárskej práce Doc. Dr. Inq. Dušan Kolářom.

V nasledujúcej kapitole sa budem zaoberať implemntáciou pomocnej aplikácie, ktorá slúži na bezpečné zadávanie a uloženie hesiel POP3 účtov.

[2] merlin.fit.vutbr.cz
[3] imap.atlas.cz
[4] eva.fit.vutbr.cz

Kapitola 7

Popis implementácie aplikácie pre bezpečné uloženie hesiel

V tejto kapitole sa budem venovať aplikácií, ktorá slúži na demonštráciu bezpečného uloženia hesiel pre aplikáciu `imapproxy`. Bude tu spomenutý aj spôsob akým táto aplkácia spolupracuje s touto aplikáciou.

7.1 Návrh aplikácie

Aplikácia je samostatná jednotka projektu. S predchádzajúcou aplikáciou zdieľa súbory `imapproxy_key`, `imapproxy.conf` a `imap.passwd`.

Po spustení aplikácie bez parametrov, aplikácia skontroluje prítomnosť súboru slúžiaceho na uloženie šifrovacieho kľúča. Ak tento súbor neexistuje, aplikácia ho vytvorí a vygeneruje náhodný kľúč, ktorý do tohto súboru uloží. Následne apikácia načíta údaje z konfiguračného súboru `imapproxy.conf` a porovnáva jednotlivé záznamy o užívateľoch a k nim prislúchajucich serverom v súbore `imap.passwd`, kde sú jednotlivé údaje uložené v tvare:
`[server] [meno užívateľa] [šifrované heslo]`.

Spôsob komunikácie s jednotlivými súbormi je zobrazený na obrázku 6.1.

7.2 Popis implementácie

Aplikácia po jej spustení prechádaza konfiguračným súborom, kde načítava potrebné údaje o mene užívateľa a servere, na ktorom je užívateľský účet uložený. Tieto informácie ukladá do poľa štruktúr. Následne prechádza do cyklu, v ktorom kontroluje prítomnosť jednotlivých údajov v súbore s heslami. Ak údaj nebol v tomto súbore nájdeny vyzýva užívateľa k zadaniu hesla pre daný účet. Toto sa opakuje pre každý účet nenájdený v súbore s heslami. Aplikácia sa pýta pre overenie správnosti na heslo dva krát. Potom vyodnotí, či dané heslo bolo správne zadané a uloží ho do spomínaného súboru.

Pre komunikáciu s užívateľom bolo vytvorené jednoduché užívateľské rozhranie s použitím nástrojov dostupných v knižnici `curses.h`.

Po zadaní hesla platného pre daný účet, aplikácia zašifruje dané heslo jednoduchým šifrovacím algoritmom. Tento algoritmus je bližšie popísaný v tejto časti práce 5.5. Šifrované heslo je uložené ako tretie slovo v riadku danom názvom servera a menom užívateľa. Heslo je uložené v hexadecimálnom tvare za použitia veľkých písmen. Pri šifrovaní bol využitý princíp prúdových symetrických šifier.

Pri spúštaní aplikácie je povolený len jeden parameter programu, ktorým je `-help`. Tento vypíše na štandardný výstup krátky popis činnosti programu.

7.3 Zhrnutie

Aplikácia bola implementovaná v prostredí operačného systému Linux (*Fedora Core 5*). Ako v predchádzajúcom prípade aj tu bol použitý prekladač *GCC* s parametrami `-std=gnu99 -pedantic -Wall -W -lcurses`. Po preklade aplikácia nesie názov `pop3passwd`.

Aplikácia slúži len ako demonštrácia uloženia hesiel pre aplikácie podobného typu ako `imapproxy`. Pre jednoduchosť svojho šifrovacieho algoritmu nie je použiteľná v aplikáciach, ktoré majú vysokú prioritu bezpečnosti.

Na tomto programe som si vyskúšal implementáciu vlastnej prúdovej šifry, taktiež, som musel vyriešiť problém bezpečného prijatia hesla od užívateľa.

Kapitola 8

Záver

Výsledkom mojej práce je implementácia aplikácie, ktorej účelom je zefektívniť a sprehľadniť poštu získanú z rôznych účtov POP3 serverov umiestnením tejto pošty na jediný účet na servere IMAP. Aplikácia svojmu užívateľovi dovoľuje previesť manipuláciu s touto poštou smerom z POP3 serverov na IMAP server v určený čas.

Súčasťou tejto práce bolo aj štúdium bezpečného uloženia citlivých informácií. Pre tento účel bol vytvorený samostatný program, ktorý demonštruje toto uloženie pre vyššie zmienenú aplikáciu. Tu som sa naučil rozpoznávať jednotlivé modely šifrovacích metód a predstavil som programovú alternatívu pre bezpečné uloženie dát v prostredí operačného systému Linux. Bola tu opísaná aj implementácia jednoduchej prúdovej šifry.

Tieto aplikácie boli vytvorené v prostredí operačného systému Linux. Úzko s touto témou súvisí aj štúdium možností a konfigurácie IMAP serverov, kde som študoval dôležté kroky pri ich nastavení v operačnom systéme Linux.

Prínosom tejto práce je aj štúdium spôsobu sťahovania pošty z POP3 serverov a porovnanie protokolov IMAP a POP3 z hľadiska náročnosti implementácie aplikácií pracujúcimi nad týmito protokolmi. Tu som nadobúdal predstavu o problémoch, ktoré musia byť riešené v týchto aplikáciach.

Takisto tu boli poukázané možné alternatívy spúšťania procesov s premennou periódou. Boli opísané ich vlastnosti a bola vybraná jedna z týchto alternatív v implementácií zmienenej aplikácie.

Hlavný prínos tejto práce vidím hlavne v poukázaní na jednu z alternatív organizácie tak často používanej služby akou je elektronická pošta.

Zoznam použitých zdrojov

[1] Mark R. Crispin. Rfc 3501. Technical report, Network Working Group, 2003.

[2] John G. Myers. Rfc 1939. Technical report, Network Working Group, 1996.

[3] WWW stránky. Asymetrické šifrovanie. http://en.wikipedia.org/wiki/Public_key.

[4] WWW stránky. Cron - wikipédia. http://en.wikipedia.org/wiki/Crontab.

[5] WWW stránky. Domovská stránka konzorcia opengroup. http://www.opengroup.org.

[6] WWW stránky. História pop3 protokolu. http://www.tcpipguide.com/free/t_POPOverviewHistoryVersionsandStandards.htm.

[7] WWW stránky. Imap na stránkach university of washington. http://www.imap.org.

[8] WWW stránky. Imap na stránkach wikipédie. http://en.wikipedia.org/wiki/IMAP.

[9] WWW stránky. Ldap na stránkach wikipédie. http://en.wikipedia.org/wiki/LDAP.

[10] WWW stránky. Open source - wikipédia. http://en.wikipedia.org/wiki/Opensource.

[11] WWW stránky. Porovnanie protokolov imap a pop3. http://www.imap.org/papers/imap.vs.pop.html.

[12] WWW stránky. Rozdelenie poštových serverov - wikipédia. http://en.wikipedia.org/wiki/List_of_mailq_servers.

[13] WWW stránky. Symetricke sifrovanie. http://en.wikipedia.org/wiki/Symmetric_cipher.

[14] WWW stránky. Test výkonnosti imap serverov. http://www.isode.com/whitepapers/mbox-benchmark.html.

[15] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall PTR, 2002. ISBN 0-13-066102-3.

[16] Andrew M. Rudoff W. Richard Stevens, Bill Fenner. *UNIX. Network Programming Volume 1, Third Edition: The Sockets Networking API*. Addison Wesley, 2003. ISBN 0-13-141155-1.

Zoznam použitých skratiek

POP3 - Post Office Protcol version 3

IMAP - Internet Message Access Protocol

RFC - Request for Comments

MIME - Multipurpose Internet Mail Extension

IETF - Internet Engeneering Task Force

IMAP4rev1 - Internet Massage Access Protocol version 4 revision 1

TCP/IP - Transmission Control Protocol/Internet Protocol

SMTP - Simple Mail Transfer Protocol

GPL - General Public Licence

LGPL - Lesser General Public Licence

LDAP - Leight-Weight Directory Access Protocol

SSL - Secure Socket Layer

TLS - Transport Layer Security

NNTP - Network News Transfer Protocol

WebMail - Web-based email

POP2 - Post Office Protocol version 2

DES - Data Encryption Standard

RSA - Rivest, Shamir, Aldeman (mená autorov)

MD5 - Message-Digest algorithm 5

SHA - Secure Hash Algorithm

PGP - Pretty Good Privacy

IDEA - International Data Encryption Algorithm

CRLF - Carriage Return, Line Feed

IAS - Sieťové aplikácie a správa sietí

IPK - Počítačové komunikácie a siete

Zoznam príloh

A Dátový nosič CD s kompletnou implementáciou implementovanej aplikácie a s programovým manuálom