

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

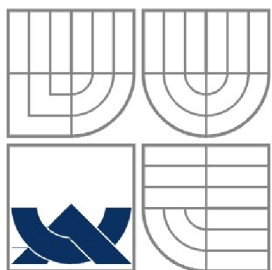
AUTOMATICKÉ STAŽENÍ E-MAILŮ PŘES WWW
ROZHRANÍ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

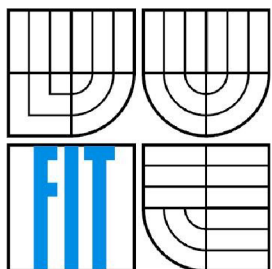
AUTOR PRÁCE
AUTHOR

MATEJ KUNA

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

AUTOMATICKÉ STAŽENÍ E-MAILŮ PŘES WWW ROZHRANÍ

AUTOMATIC E-MAIL DOWNLOAD THROUGH WWW INTERFACE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MATEJ KUNA

VEDOUCÍ PRÁCE

SUPERVISOR

Doc. Dr. Ing. DUŠAN KOLÁŘ

BRNO 2008

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů

Akademický rok 2007/2008

Zadání bakalářské práce

Řešitel: **Kuna Matej**

Obor: Informační technologie

Téma: **Automatické stažení e-mailů přes WWW rozhraní**

Kategorie: Web

Pokyny:

1. Prostudujte různé bezplatné systémy pro poskytování e-mailu přes WWW rozhraní.
2. Analyzujte shodné a rozdílné prvky pro přístup k poště (od textu, přes přílohy po hlavičky). Soustředte se na analyzovatelnost na HTML úrovni.
3. Navrhněte vhodně parametrizovatelnou a jednoduše konfigurovatelnou aplikaci pro automatické stahování e-mailů přes WWW rozhraní.
4. Implementujte navržený nástroj, či systém nástrojů.
5. Ověřte správnou funkčnost alespoň na těchto 3 různých poskytovatelích e-mailu: operamail.com, sms.cz, quick.cz.
6. Zhodnoťte návrh a implementaci. Diskutujte případné nedostatky a možná rozšíření.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- První 3 body zadání. Neparametrický prototyp k bodu 4.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

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

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

Vedoucí: **Kolář Dušan, doc. Dr. Ing.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2007

Datum odevzdání: 14. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2

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: **Matej Kuna**

Id studenta: 78902

Bytem: Zvolenská 7, 949 11 Klokočina

Narozen: 09. 07. 1986, Nitra

(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: Automatické stažení e-mailů přes WWW rozhraní

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 tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel

.....
Autor

Abstrakt

V tejto práci sa zaoberám ukladáním elektronickej pošty zo serverov, ktoré ponúkajú prístup k e-mailovej schránke prostredníctvom internetového prehliadača. Analyzujem a porovnávam niekoľko takýchto serverov. V konečnej fáze vytváram program, ktorý by dokázal pristupovať k e-mailovej schránke bez asistencie užívateľa.

Kľúčové slová

elektronická pošta, webmail, e-mailový klient, Python

Abstract

In this thesis I deal with downloading e-mails from servers which offer access to the inbox via a web browser. I analyze and compare several web-based e-mail servers. In the final stage I have created program, which is able to access to the inbox without user assistance.

Keywords

e-mail, web-based email, e-mail client, Python

Citácia

Matej Kuna: Automatické stažení e-mailů přes WWW rozhraní , bakalárska práca, Brno, FIT VUT v Brne, 2008

Automatické stažení e-mailů přes WWW rozhraní

Prehlásenie

Vyhlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Doc. Dr. Ing. Dušana Koláňa. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Matej Kuna
11.3.2008

Pod'akovanie

Chcel by som sa poďakovať môjmu vedúcemu Doc. Dr. Ing. Dušanovi Kolárovi, ktorý ma pri riešení práce viedol správnym smerom. Rovnako nemôžem zabudnúť na ľudí z IRC kanálu #python na serveri FreeNode, ktorí mi pomohli pri problémoch, s ktorými som sa stretol pri písaní programu.

© Matej Kuna, 2008.

Táto práca vznikla ako školské dielo na Vysokom učení technickom v Brne, Fakulte informačných technológií. Práca je chránená autorským zákonom a jej použitie bez udelenia oprávnenia autorom je nezákonné, s výnimkou zákonom definovaných prípadov.

Obsah

Obsah.....	1
1 Úvod.....	2
2 Rôzne protokoly pre prístup k elektronickej pošte.....	3
2.1 Formát e-mailu.....	3
2.2 Definícia protokolu.....	6
2.3 Protokol POP3.....	6
2.4 Protokol IMAP.....	7
2.5 Protokol HTTP.....	8
2.5.1 HTTP hlavička.....	9
2.6 Webmail.....	10
3 Analýza systémov poskytujúcich e-mail.....	12
3.1 Základné informácie o serveroch.....	12
3.2 Analýza WWW rozhrania.....	14
4 Návrh aplikácie.....	16
5 Implementácia aplikácie.....	18
5.1 Modularita.....	19
5.1.1 Modul pre nastavenie aplikácie.....	19
5.1.2 Komunikačný modul.....	20
5.1.3 Modul pre správu súborov.....	21
5.1.4 Modul pre odosielanie e-mailov (neimplementované).....	22
5.2 Výsledná aplikácia.....	22
6 Výsledky.....	23
7 Záver.....	24
7.1 Ďalší vývoj aplikácie.....	24
Literatúra.....	26
Zoznam príloh.....	28
A CD-ROM.....	29

1 Úvod

Aj keď nie sú známe žiadne oficiálne štatistiky, pokojne sa môže tvrdiť, že najväčší tok dát v rámci siete Internet tvorí komunikácia cez HTTP protokol. Inak povedané prezeranie WWW stránok. Potrebný program, internetový prehliadač, je v dnešnej dobe súčasťou každého operačného systému. Pre užívateľa s menším prehľadom v oblasti internetových služieb je HTTP protokol najpriateľnejší, pretože internetový prehliadač nevyžaduje žiadnu konfiguráciu¹. To je dôvod, prečo sa čoraz častejšie môžeme stretnúť s tým, že aj niektoré iné sieťové protokoly sú prístupné cez WWW rozhranie, teda HTTP protokol. Takto je existencia iného protokolu pre konečného užívateľa nepodstatná a vôbec o nej nevie. Tak odpadá nutnosť mať nainštalovaný špeciálny program podporujúci daný protokol ako i jeho prípadná konfigurácia.

V mojej práci sa zaoberám elektronickou poštou. Na jej čítanie boli navrhnuté protokoly POP3 a IMAP. Takmer každý e-mailový server umožňuje prístup k pošte aj cez WWW rozhranie. Bohužiaľ, niekedy to je jediný spôsob ako čítať a posilať elektronickú poшту. Týka sa to iba voľne dostupných e-mailových serverov a práve nimi sa zaoberám v tejto práci. Cieľom mojej práce je vyvinúť nástroj, ktorý by bol schopný ukladať e-maily pomocou WWW rozhrania bez asistencie užívateľa.

Druhá kapitola stručne definuje pojem protokol a zaoberá sa rôznym prístupom k elektronickej pošte. Sú v nej porovnané vyššie spomínané protokoly POP3 a IMAP. Záver kapitoly je venovaný prístupu k e-mailu cez WWW rozhranie a HTTP protokol.

V tretej kapitole opisujem voľne dostupné e-mailové servery. Vybral som si niekoľko serverov, ktoré hlbšie analyzujem a skúmam. Výsledky ukazujú, že servery naozaj ponúkajú bezplatne iba prístup cez webové rozhranie. Ďalšie protokoly bývajú spoplatnené, takže malo zmysel implementovať aplikáciu umožňujúcu automatický prístup k pošte.

Kapitola číslo 4 je venovaná návrhu aplikácie. Nadväzuje na predchádzajúcu analýzu a zaoberá sa tým, čo všetko musí program splňať, aby mohol automaticky komunikovať s e-mailovým serverom cez HTTP protokol a ukladať prípadné e-maily.

Predposledná kapitola podáva čitateľovi obraz o tom, ako bola aplikácia implementovaná. Je tu uvedená schéma rozdelenia programu do samostatných modulov. Každému modulu je venovaná samostatná podkapitola, ktorá uvádza potrebné informácie.

Záver práce patrí celkovému hodnoteniu. Diskutujem tu o nedostatkoch, ale aj výsledkoch implementácie. Stručne som zhrnul informácie o tom, akým smerom by mohol napredovať prípadný ďalší vývoj práce.

1 Ako každý program tak aj internetový prehliadač má nejakú formu konfigurácie. Ale v porovnaní s poštovým klientom je možné s prehliadačom pracovať hneď po nainštalovaní bez zásahu do konfigurácie.

2 Rôzne protokoly pre prístup k elektronickej pošte

Prvá podkapitola sa zaoberá formátom e-mailu. Za ňou je uvedená definícia protokolu a vysvetlený presný význam tohto termínu. V nasledujúcich dvoch podkapitolách je krátky popis protokolov slúžiacich na čítanie elektronickej pošty. Teda protokoly POP3 a IMAP. Vôbec sa nebudem zaoberať konkrétnymi príkazmi týchto protokolov. Rovnako bude vynechaný protokol SMTP, ktorý slúži na zasielanie elektronickej správ. Záver tejto kapitoly je venovaný protokolu HTTP a úvaha o tom, ako ho je možné využiť na čítanie e-mailov.

Elektronická pošta býva umiestnená na serveroch s permanentným pripojením k Internetu. Tie ponúkajú prostriedky klientským programom pre prístup k pošte. Klientský program dokáže komunikovať so serverom cez protokol. Pojem *server* sa vzťahuje na objekt, ktorý ponúka určitú službu. Tú využíva *klient* a počas vzájomnej komunikácie sa dorozumievajú cez stanovený *protokol*.

2.1 Formát e-mailu

E-mail má definovaný formát podľa viacerých dokumentov. Jeden z tých starších [15] hovorí, že správu tvoria znaky ASCII tabuľky² v rozpätí hodnôt od 1 po 127 vrátane. Znaková sada má označenie aj US-ASCII (sem patrí aj prvý nultý znak). Správa je rozdelená do riadkov. Riadok pozostáva zo znakov a je ukončený znakmi CRLF³. Existujú aj určité obmedzenia čo sa týka dĺžky jedného riadku. Správa je tvorená hlavičkou, za ňou nasleduje telo. Hlavička a telo správy sú od seba oddelené prázdny riadkom. Z popisu je zrejmé, že syntax dokáže popísať iba textové správy. V tomto formáte neexistuje žiadne opatrenie pre prenos obrazových, zvukových a iných druhov údajov cez elektronickej poštu. Vzťahuje sa na e-maily prenášané po sieti a nerieši akým spôsobom má byť správa uložená lokálne. Pre lokálne uloženie neexistuje žiadny štandard, ale navrhnutý formát sa často používa aj za týmto účelom.

Hlavička správy sa skladá z polí s nasledovným formátom. Pole sa začína názvom (*field name*), za ktorým je dvojbodka a nasleduje hodnota tohto poľa (*field body*). Nie je definované žiadne poradie v akom majú názvy polí za sebou nasledovať, teda sa môžu vyskytovať v ľubovoľnom poradí. Môže nastať aj situácia, že polia sa počas prenosu po sieti preskupia. Niektoré polia sa môžu vyskytnúť iba raz, iné aj viackrát. V tabuľke 2.1 je viacero príkladov.

² ASCII tabuľka – 8 bitový kód reprezentujúci textové znaky, čísla, interpunkčné znamienka a špeciálne znaky
³ CRLF – carriage return + line feed, dvojica znakov z ASCII tabuľky 0x13 (CR) a 0x10 (LF) reprezentujúca nový riadok

Telo správy je len obyčajný text. Vzťahujú sa naň obmedzenia vyplývajúce z celej správy a jedného riadku správy. Telo správy je len voliteľné a predstavuje objekt, ktorý má byť doručený adresátovi.

Popísaný formát upravujú a rozširujú ďalšie dokumenty [5, 6, 9] so spoločným názvom MIME – *Multipurpose Internet Mail Extensions*. Správy v novšom formáte môžu mať telo alebo aj hlavičku v inej znakovnej sade ako US-ASCII. Bol pridaný rozširujúci popis rôznych formátov pre telo, ktoré neobsahuje text. Je dovolené prenášať aj viacero objektov súčasne. Táto správa je nazývaná ako multimedialná správa. Spomínané rozšírenie je zabezpečené pridaním nových polí do hlavičky e-mailu. Je to rozšírenie, ktoré berie ohľad na spätnú kompatibilitu, takže správa v staršom formáte je úplne správna aj vzhľadom na pravidlá novšieho formátu.

Názov	Použitie
From	From: Matej Kuna <mkuna@wmw.sk>
Sender	Sender: Ewi <secretary@wmw.sk>
Reply-To	Reply-To: "Matej: Home Account" <matej@home.sk>
To	To: Matej Kuna <mkuna@wmw.sk>
Cc	Cc: <boss@example.sk>, <syservice@example.sk>
Bcc	Bcc: Ewi <secretary@wmw.sk>
Date	Date: Tue, 11 Mar 2008 17:08:37 +0100
Subject	Subject: Pozdrav
MIME-Version	MIME-Version: 1.0
Content-Type	Content-Type: text/plain; charset="iso-8859-2"
Content-Transfer-Encoding	Content-Transfer-Encoding: 7bit

Tabuľka 2.1: polia v hlavičke e-mailu

From špecifikuje čiarkami oddelený zoznam jednej alebo viacerých adries e-mailových schránok.

Musí ho obsahovať každá správa, pretože toto pole je povinné. Určuje autora alebo prípadne autorov správy. To znamená osobu alebo program zodpovedný za vytvorenie správy. Ak je poštových schránok uvedených viac ako jedna, tak musí byť uvedené aj pole **Sender**.

Sender nesie v sebe práve jednu adresu. Označuje toho, kto je zodpovedný za odoslanie správy. To znamená, že ak je autor a odosielateľ správy rôzny, budú uvedené obe príslušné polia.

Reply-To obsahuje adresu e-mailovej schránky, kam má prísť prípadná odpoveď na správu. Ak toto pole uvedené nie je, odpoveď príde na adresu uvedenú v poli **From**.

Date je druhou a poslednou povinnou položkou. Určuje dátum a čas, kedy pisateľ označil správu za dokončenú a pripravenú na doručenie. Napríklad to môže byť dátum a čas kedy užívateľ vo svojom programe stlačil tlačidlo symbolizujúce odoslanie správy. V každom prípade táto informácia nemá udávať čas, kedy bola správa prenášaná, ale čas kedy bola tvorcom napísaná a

dokončená. Tvorcom nemusí byť len človek, ale môže ísť aj o počítačový program. Ak napríklad užívateľ nie je pripojený k Internetu, tak môže správu napísať a uložiť na neskoršie odoslanie. Dátum a čas by potom mal označovať skutočnú dobu vytvorenia a nie dobu, kedy bola správa odoslaná.

Subject síce nie je povinný, ale má ho väčšina správ. Nesie v sebe krátku charakteristiku obsahu správy. Ak ide o odpoveď na nejakú predchádzajúcu správu, tak sa môže začínať reťazcom „RE:“.

Comment obsahuje ľubovoľný komentár k telu e-mailu.

Keywords je čiarkami oddelený zoznam dôležitých slov alebo fráz, ktoré môžu byť pre adresáta dôležité.

To obsahuje zoznam adres adresátov, ktorým je správa primárne určená.

CC udáva zoznam ďalších adres, kam bude správa poslaná, aj keď im nie je primárne určená.

BCC obsahuje adresy adresátov, ktorých adresy nie sú pre ostatných prijímateľov podstatné. Je niekoľko spôsobov ako sa s touto informáciou zaobchádza. Tento riadok sa môže odstrániť a kópia správy sa pošle všetkým existujúcim adresátom. Alebo sa pošle kópia správy všetkým adresátom uvedených v BCC (aj s týmto riadkom), následne sa tento riadok odstráni a pošle sa druhá kópia zvyšným adresátom (bez tohto riadku).

MIME-Version je použitá na definovanie verzie tela elektronickej správy. Bola zavedená až v novšom MIME štandarde a jej prítomnosť značí, že správa bola podľa neho komponovaná. Zvyšné polia sa rovnako týkajú novšieho štandardu.

Content-Type – účelom je podať dostatočnú informáciu o tele správy tak, aby mohol adresátov program vybrať vhodný mechanizmus pre prezentáciu obsahu. Špecifikuje podstatu tela danú typom a podtypom. Pre určité podtypy sa môže objaviť množina parametrov, ktoré ho bližšie došpecifikujú.

Content-Transfer-Encoding určuje typ použitého kódovania. Ak má správa niest' aj binárne údaje alebo len text s bohatšou znakovou sadou ako US-ASCII, musí existovať metóda, ktorá by umožňovala tieto údaje prekódovať do US-ASCII. Adresátov klientský program je na základe tohto údaju schopný správu spätne dekodovať. Ak by sa poslala správa s väčšou znakovou sadou ako US-ASCII (viac ako 7 bitov pre jeden znak) bez prekódovania, jej prenos by mohol zlyhať.

2.2 Definícia protokolu

Protokol je štandard, ktorý umožňuje spojenie a komunikáciu medzi dvomi bodmi v počítačovej sieti. Každý protokol definuje svoje príkazy, ich syntax a sémantiku. Obyčajne protokol určuje ako sa ustanovuje komunikácia, samotnú komunikáciu a jej korektné ukončenie. Protokol môže byť implementovaný hardwarovo aj softwarovo. Ďalší popis sa bude týkať softwarových protokolov, ktoré pracujú na aplikačnej vrstve modelu architektúry TCP/IP.

2.3 Protokol POP3

POP3 [11] je skratka od *Post Office Protocol - Version 3*. Cieľom tohto protokolu je klientovi umožniť vybratie elektronickej pošty zo schránky umiestnenej na serveri. To znamená stiahnutie pošty zo servera, kde je uložená, do počítača užívateľa. Nebol vytvorený tak, aby poskytoval širokú škálu operácií nad elektronickými správami na strane e-mailového serveru. Základná implementácia ponúka iba dva príkazy vzťahujúce sa k jednému e-mailu. Je to jeho zobrazenie a vymazanie.

Jedná sa o aplikačný protokol očakávajúci TCP spojenie. Server poskytujúci prístup k pošte cez protokol POP3 má túto službu spustenú na porte 110. Ak chce klient používať službu, so serverom ustanoví spojenie. Ak sa toto podarí, server pošle klientovi uvítaciu správu. Potom si klient a server vymieňajú príkazy a príslušné reakcie, pokiaľ nie je spojenie uzatvorené alebo sa nepreruší.

POP3 príkazy pozostávajú z kľúčových slov, ktoré nerozlišujú veľkosť písmen. Môžu obsahovať jeden a viac argumentov. Každý príkaz je ukončený znakmi CRLF. Kľúčové slová a argumenty pozostávajú z viditeľných znakov (nie medzera, prechod na novú stránku a podobne). Kľúčové slová a argumenty sú od seba oddelené medzerou. Kľúčové slovo je dlhé 3 alebo 4 znaky, argument môže mať dĺžku až 40 znakov.

Odpoveď servera na príkaz pozostáva z indikátora stavu, za ktorým môžu nasledovať ďalšie informácie. Odpoveď môže mať aj viac riadkov a je dohodnutým spôsobom ukončená. V súčasnosti existujú dva indikátory stavu. Pozitívny indikátor +OK. Reakcia na príkaz je kladná. Negatívny indikátor má tvar -ERR a značí negatívnu odpoveď na predchádzajúci príkaz.

Komunikácia medzi klientom a serverom prechádza cez viacero stavov. Po ustanovení spojenia a odoslania uvítacej správy klientovi komunikácia prechádza do stavu označovaného *authorization*. V tomto stave sa musí klient POP3 serveru identifikovať. Existuje viacero spôsobov autorizácie. Nie je presne definované, ktorý spôsob musí server podporovať, najmenej jeden však musí byť podporovaný.

Ak sa klient úspešne identifikuje, komunikácia nadobudne stav *transaction*. Server opatrí schránku zámkom. Je to prevencia pred akoukoľvek modifikáciou správ pred nadobudnutím

konečného stavu. V tomto stave je možné opakovane používať príkazy pre čítanie a mazanie konkrétnych správ. Ďalej sú k dispozícii príkazy zobrazujúce počet a zoznam správ.

Keď klient zašle na server príkaz, ktorým ukončuje spojenie, tak prichádza stav s názvom *update*. Server odstráni všetky správy, ktoré boli za týmto účelom označené v predošlom stave. Uvoľní zo schránky zámok a TCP spojenie uzatvorí.

2.4 Protokol IMAP

IMAP [2] je skratka od *Internet Message Access Protocol*. Verzia 4 (IMAP4) umožňuje prístup k elektronickej pošte a jej manipuláciu na strane serveru. Dovoľuje vytvárať pre e-maily schránky priamo na serveri, s ktorými je možné prevádzať podobné funkcie ako s priečkami na pevnom disku. Je ich možné vytvárať, mazať a premenovávať.

IMAP server má túto službu spustenú na TCP porte 143. Komunikácia spočíva v ustanovení sieťového spojenia medzi serverom a klientom, za ktorým nasleduje uvítacia správa od serveru. Potom môže aplikácia klienta a servera medzi sebou vzájomne komunikovať prostredníctvom správ. Správy majú formu riadkov, teda reťazcov a sú ukončené znakmi CRLF.

Ak bolo ustanovené spojenie medzi klientom a serverom, tak je komunikácia v jednom zo štyroch stavov. Väčšina príkazov je dostupných iba v určitom stave. Po poslaní uvítacej správy klientovi od servera je komunikácia v stave *not authenticated*. Klient sa musí identifikovať. V tomto stave je väčšina príkazov zakázaná. V ďalšom stave s menom *authenticated* je klient identifikovaný a musí si zvoliť e-mailovú schránku. Až potom budú povolené príkazy týkajúce sa e-mailových správ.

Po úspešnom zvolení e-mailovej schránky komunikácia prechádza do stavu *selected*, kde môže prevádzať operácie nad správami. V poslednom stave *logout* je spojenie ukončené. Do tohto stavu sa zvyčajne komunikácia dostane, keď server obdrží od klienta príkaz odhlásenia.

E-mailové správy uložené na IMAP serveri majú niekoľko atribútov. Prvým z nich je jedinečné číslo, pod ktorým je každá správa prístupná. Ďalej môže byť správa označená viacerými príznakmi. Tie napríklad určujú, že správa bola prečítaná, bolo na ňu odpovedané, bola označená ako dôležitá alebo bola označená ako rozpísaná. Nie sú to všetky príznaky a server môže dovoliť klientovi definovať vlastné. E-mailu sa príznak nastavuje jeho pridaním do zoznamu a zaniká odobratím. Príznak môže byť trvalý alebo platný len po dobu pripojenia k serveru.

E-mail má aj svoj vnútorný dátum a čas. Tento údaj určuje kedy bola správa prijatá. K atribútom e-mailu patrí ešte veľkosť správy. Atribúty uzatvára hlavička a telo správy, tak ako je to popísané v kapitole 2.1. Z toho plynie možnosť získať len samotnú hlavičku alebo samotné telo správy.

Tento protokol v porovnaní s POP3 protokolom prináša nové funkcie. Doteraz nebola zmienka o tom, že bez problémov podporuje prístup k tej iste mailovej schránke súčasne cez viacero klientov. Klientské aplikácie majú v sebe zabudované mechanizmy, ktoré umožňujú rozpoznať zmenu, ktorá vznikla na serveri inou, súčasne pripojenou aplikáciou. Server ponúka možnosť vyhľadávať e-maily, ktoré spĺňajú určité kritériá. Nie je potrebné ich pred vyhľadávaním ukladať na disk. Konkrétnymi príkazmi klienta a príslušnými odpoveďami servera sa už nebudem zaoberať.

2.5 Protokol HTTP

HTTP [3] je skratka od *Hypertext Transfer Protocol*. Bol navrhnutý pre distribuované a spolupracujúce dokumenty, ktoré obsahujú grafiku, audio, video a text. Dokument (dokumenty) spravidla obsahuje aj odkazy, ktoré v ňom rušia lineárnosť. Prvá verzia protokolu bola HTTP/0.9 a používala sa od roku 1990. Bola jednoduchá a určená pre distribúciu nijak neupravených dát cez sieť Internet. Veria HTTP/1.0 protokol vylepšila. Umožnila pripojiť k prenášaným dátam metainformácie a modifikátory. Zaviedla MIME formát prenášaných správ. Posledná a dnes používaná verzia je HTTP/1.1, ktorá priniesla ešte viac vylepšenú funkcionálnosť.

HTTP server načúva na TCP porte 80. Tento protokol je typu požiadavka/odpoveď. Komunikáciu zahajuje klient zaslaním požiadavky na server. Pozostáva z požiadavok klienta a odpovedí serveru. Správy reprezentujúce požiadavky aj odpovede majú rovnaký formát. Je takmer identický s formátom e-mailu popisovaného v kapitole 2.1. HTTP správa začína štartovacím riadkom, pokračuje hlavičkou a končí telom. Hlavička a telo sú od seba oddelené CRLF. Hlavičku a ani telo správa obsahovať nemusí. Týka sa to predovšetkým požiadavky, pretože odpoveď bez obsahu nenesie žiadnu užitočnú informáciu. Z hľadiska definície protokolu je však aj takáto prázdna (musí mať stavový kód) odpoveď povolená.

Požiadavka od klienta na server na prvom riadku pojíma *metódu*, *identifikátor zdroja* a použitú *verziu HTTP* protokolu. Každý server musí mať implementované *metódy* GET a HEAD. Bude stačiť tvrdenie, že metóda GET má všeobecné použitie a je najčastejšie používaná. Ak klient použije HEAD, tak žiada server iba o zaslanie samotnej hlavičky HTTP odpovede. Ďalšia používaná metóda je POST. Môže sa použiť na spracovanie dát odoslaných cez HTML formulár na server. Existuje ešte 5 iných metód, ale ich použitie je ojedinelé. Druhou povinnou položkou požiadavky je *identifikátor zdroja* – ten jednoznačne určuje objekt, na ktorý je požiadavka smerovaná a klient ho chce získať. *Verzia HTTP* protokolu býva dnes takmer vždy HTTP/1.1. Všetky tri uvedené položky požiadavky sú v jednom riadku a ten je od prípadnej hlavičky oddelený prázdny riadok. V hlavičke požiadavky môže klient poslať na server prídavné ďalšie informácie o požiadavke alebo o sebe samotnom.

Po doručení a spracovaní správy s požiadavkou server odpovie HTTP správou. Na prvom riadku je verzia protokolu (pretože sa môže od verzie v požiadavke líšiť), číselný stavový kód a krátka textová prezentácia stavového kódu. Stavový kód je trojciferný. Prvá číslica udáva kategóriu odpovede (tabuľka 2.2 [4]), posledné dvojčíslo presne určuje jej typ.

Stavový kód	Význam
1xx	Informácia – požiadavka prijatá, pokračovanie v procese.
2xx	Úspech – požiadavka bola úspešne doručená a prijatá
3xx	Presmerovanie – musí byť vykonaná ďalšia akcia pred dokončením požiadavky
4xx	Chyba klienta – požiadavka má zlý tvar alebo nemôže byť uspokojená
5xx	Chyba serveru – server zlyhal pri spracovaní zdanlivo správnej požiadavky

Tabuľka 2.2: stavové kódy odpovede HTTP serveru

2.5.1 HTTP hlavička

Accept umožňuje špecifikovať určité druhy médií pomocou typu a podtypu, ktoré sú prijateľné ako odpoveď na predchádzajúcu požiadavku. Pri vynechaní sa predpokladá, že klient akceptuje všetky typy. Ak server nemá požadovaný typ k dispozícii, mal by vrátiť stavový kód s číslom 406.

Accept-Charset udáva znakovú sadu očakávanú v odpovedi. Ak pole nie je uvedené, je zvolená predvolená znaková sada. Ak server nedokáže poslať požadovanú odpoveď, mal by vrátiť stavový kód 406.

Accept-Encoding umožňuje požadovať cieľový objekt v určitom kódovaní. Týka sa to predovšetkým použitia kompresie takým spôsobom, aby cieľ po použití reverzného algoritmu nestratil žiadnu informačnú hodnotu objektu. Pri nemožnosti poslať odpoveď v požadovanom kódovaní, by stavový kód mal byť 406.

Accept-Language sa vzťahuje na prirodzený jazyk požadovaného zdroja.

Host spolu s prvým riadkom požiadavky určuje adresu požiadavky. V najnovšej verzii protokolu (HTTP 1.1) musí byť vždy uvedený. Inak bude mať výsledná odpoveď stavový kód 400.

User-Agent obsahuje informáciu o programe, ktorý uskutočňuje požiadavku. Má len štatistický význam alebo dovoľuje rozpoznanie programu. Táto hodnota by mala byť v požiadavke zahrnutá.

Location je uvedené pri presmerovaní a určuje novú adresu požiadavky, ktorá sa od pôvodnej odlišuje. Pre jej dokončenie a získanie odpovede je potrebné požadovať novú adresu. Obyčajne býva toto pole uvedené spolu s odpoveďou so stavovým kódom 3xx.

2.6 Webmail

Webmail je služba poskytujúca e-mailovú schránku primárne cez WWW rozhranie. V súčasnosti existuje veľké množstvo voľne dostupných serverov, ktoré ponúkajú e-mailovú schránku bez akýchkoľvek poplatkov. Pre niektorých užívateľov je toto jediná možnosť ako získať e-mailovú adresu pre komunikáciu. Odosielanie a čítanie správ je možné cez WWW rozhranie prostredníctvom HTTP protokolu. Klient na čítanie a odosielanie správ je rovnaký ako program, ktorým užívateľ prístupuje k WWW stránkam, inak povedané, klientský program tvorí internetový prehliadač a webové rozhranie serveru.

Niektoré servery umožňujú prístup k prijatej pošte aj cez POP3 prípadne IMAP protokol. Nemusí to však byť pravidlom. Obyčajne servery spomínané protokoly podporujú, ich využitie je spoplatnené. Vo zvyšku tejto práce sa budem venovať len čítaniu pošty cez protokol HTTP.

Webové rozhranie, ktoré umožňuje prístup užívateľom k ich pošte je na každom serveri unikátne. Z toho plynie fakt, že prístup k dvom e-mailovým schránkam na rôznych serveroch sa môže výrazne odlišovať. Existujú aj voľne dostupné webové aplikácie pre prístup k pošte. Nebolo však zistené, že by ich nejaký verejne dostupný server používal. Ukážka webového rozhrania je na obrázku 2.1. Je vytvorené v nejakom značkovacom jazyku [17] a na definíciu vzhľadu využíva CSS [1]. Čo sa značkovacích jazykov týka, tak používané sú HTML a XHTML. Oba majú do značnej miery príbuznú syntax. Vo veľkom rozsahu sa spolieha aj na možnosti skriptovacieho jazyka na strane klienta – JavaScript [10].

Keďže užívateľ na prístup k pošte využíva len internetový prehliadač komunikujúci cez HTTP protokol, neznamená to, že protokoly POP3 a IMAP nemajú žiadny význam. Konkrétna implementácia nie je dostupná, ale je dosť pravdepodobné, že HTTP server spomínané protokoly skryto využíva. Webové rozhranie je v tom prípade iba sprostredkovateľ komunikácie medzi užívateľom a skutočným úložiskom e-mailov.

Elektronická správa ako taká stráca svoj formát. Je reprezentovaná v značkovacom jazyku daného rozhrania spolu s jej hlavičkou. Hlavička nemusí byť úplná a niektoré položky môžu chýbať, pretože primárnou úlohou serveru je užívateľovi prezentovať obsah správy. Celá správa je vsadená do webového rozhrania a spolu s ním tvorí jeden celok. Niekedy existuje možnosť zobrazit' iba samotnú správu vo formáte opísanom v kapitole 2.1 v celom rozsahu.

OPERA Web Mail
FREE WEBMAIL SERVICE

[CHECK MAIL](#)
[FOLDERS](#)
[COMPOSE](#)
[CONTACTS](#)
[OPTIONS](#)
[SMS](#)
[LOGOUT](#)
[PREMIUM SERVICE](#)

You are using 72.19% of your mail quota.

INBOX - 10 Message(s), 0 New [Get External Mail](#) | [Empty Trash](#)

View All Messages Showing 1 to 10 of 10 Please select a folder

<input type="checkbox"/>	<input type="checkbox"/>	From	Subject	Date >>	Size
<input type="checkbox"/>	<input type="checkbox"/>	Loforte Seager	coagulase	1 Apr 2008	3k
<input type="checkbox"/>	<input type="checkbox"/>	Tetley Crookston	tartar	31 Mar 2008	3k
<input type="checkbox"/>	<input type="checkbox"/>	Matej Kuna	adadg	25 Mar 2008	1k
<input type="checkbox"/>	<input type="checkbox"/>	Matej Kuna	vela priloh	11 Mar 2008	225k
<input type="checkbox"/>	<input type="checkbox"/>	Ondrej Domotor	priloha *.php *.jpg *.html	5 Mar 2008	52k
<input type="checkbox"/>	<input type="checkbox"/>	Ondrej Domotor	priloha funbox :)	5 Mar 2008	936k
<input type="checkbox"/>	<input type="checkbox"/>	wizard white	list	5 Mar 2008	1k
<input type="checkbox"/>	<input type="checkbox"/>	Matej Kuna	priloha	26 Jan 2008	49k
<input type="checkbox"/>	<input type="checkbox"/>	Fannie Hooks	RE: Bar-Ray Drino Priammid	31 Dec 2007	2k
<input type="checkbox"/>	<input type="checkbox"/>	operamail.com Member Services	Welcome to your Opera Web Mail e-mail account	11 Nov 2007	1k

=Low Priority =High Priority =Flagged

Obrázok 2.1: webové rozhranie serveru www.operamail.com

3 Analýza systémov poskytujúcich e-mail

Zaoberal som sa niekoľkými servermi poskytujúcimi e-mailovú schránku cez WWW rozhranie. Vybral som si dva zahraničné servery, ale aj tri české e-mailové servery. Na každom serveri som si zriadil účet. Ich adresy spolu s výsledkami analýzy sú zhrnuté v tabuľkách v prvej podkapitole. Niektoré z výsledkov som použil pri implementácii programu. V závere kapitoly sa nachádza dôležité zhrnutie celej analýzy.

3.1 Základné informácie o serveroch

Najprv boli zhromaždené základné informácie o jednotlivých serveroch. Sem patrí ponúkaná veľkosť e-mailovej schránky a možnosť bezplatného prístupu k pošte cez protokoly POP3 a IMAP. Nakoniec bolo prevedené prihlásenia sa k jednotlivým účtom, otvorenie schránky s doručenou poštou a získaný obsah sa predal validátoru [16]. Posledný stĺpec v tabuľke 3.1 ukazuje počet chýb, ktoré obsahovali. Síce počet chýb nie je až tak dôležitý ako ich typ. V každom prípade nešlo o závažné chyby, ktoré by vážnejšie narušali (X)HTML štruktúru dokumentu.

adresa serveru	veľkosť schránky	POP3 a IMAP	validátor
mail.google.com	viac ako 5000 MB	áno	28
www.operamail.com	3 MB	nie	19
www.centrum.cz	20 MB	POP3	11
www.o2active.cz	100 MB	nezistené	10
email.sms.cz	100 MB	nie	182

Tabuľka 3.1: základné informácie o serveroch

Ďalej sa analyzovalo aké vlastnosti musí mať internetový prehliadač, aby bolo možné jeho prostredníctvom klasickým spôsobom prístupit' k pošte. Najprv bolo realizované prihlásenie sa k účtom cez internetový prehliadač Lynx [7] s predvolenými nastaveniami. Je to textový internetový prehliadač. Funkčnosť spomínaných serverov v tomto prehliadači bola testovaná z toho dôvodu, lebo je dosť pravdepodobné, že vo fáze implementácie bude moja aplikácia pracovať s HTML kódom podobnými metódami ako tento prehliadač. Tabuľka 3.2 znázorňuje, či bolo prihlásenie úspešné.

Potom sa funkčnosť testovala klasickým internetovým prehliadačom a skúmala sa potreba mať povolené ukladanie cookies⁴ [8] a nutnosť mať zapnutú podporu jazyka JavaScript [10].

Cookie predstavuje malý objem dát, ktoré si môže uložiť každý HTTP server. O samotné ukladanie sa stará internetový prehliadač. Cookie má vždy svoje meno, hodnotu, dobu a rozsah platnosti a server, ku ktorému prislúcha. Rozsah platnosti je daný adresárom existujúcim na serveri a hodnota cookie existuje iba ak je v prehliadači otvorený daný adresár alebo ľubovoľný jeho podadresár. Ak nie je adresár uvedený, tak hodnota je dostupná globálne na celom serveri. Je možné uviesť aj časový údaj, kedy už hodnota nemá byť platná a prehliadač ju má zmazať. Niekedy sa tento údaj nastavuje do minulosti pre nejaké už uložené meno cookie, čo zapríčini zmazanie existujúcej hodnoty. Jedna hodnota môže mať veľkosť maximálne 4kB.

adresa serveru	Lynx	Cookies (blokované)	JavaScript (vypnutý)
mail.google.com	áno	nie	áno *
www.operamail.com	áno	nie	áno
www.centrum.cz	áno	áno	áno *
www.o2active.cz	nie	áno	nie
email.sms.cz	nie	nie	áno

Tabuľka 3.2: vlastnosti internetového prehliadača

Z výsledku (tabuľka 3.2) je vidieť, že väčšina serverov potrebuje mať v internetovom prehliadači povolené cookies. Je dosť pravdepodobné, že aj servery, na ktoré sa podarilo prihlásiť bez cookies by pri zapnutej podpore cookies použili.

Veľký problém sa ukázal jazyk JavaScript. Server *www.operamail.com* mal značne obmedzenú funkčnosť. Nebolo napríklad možné e-mail vymazať, ale zobrazit' sa dal. Preto som tento server označil tak, že v teste uspel. Kurióznym prípadom nastal na serveri *email.sms.cz*. Udalosti jazyka JavaScript využíva prihlasovací formulár. Pri jeho vypnutí bolo vidieť písané heslo, ktoré by malo byť za normálnych okolností kryté hviezdičkami. Z toho vyplýva, že heslo sa nevkladalo do (X)HTML elementu, ktorý bol na tento účel určený.

Vypnutie podpory jazyka JavaScript najlepšie zvládli servery označené hviezdičkou (*). Funkčnosť nebola žiadnym spôsobom obmedzená, až na ten fakt, že užívateľa informujú, že táto podpora je vypnutá.

Posledná časť analýzy sa týka už samotného kódu značkovacieho jazyka. Presná verzia a ani to, či sa jedná o HTML alebo XHTML nie je dôležité. Táto informácia je podstatná hlavne pre internetový prehliadač, aby vedel obsah správne vykresliť. Serverom *www.o2active.cz* sa už viac

4 cookies – neexistuje vhodný preklad tohto anglického termínu, jednotné číslo je cookie

nebudem zaoberať. Keďže udržiavať stav HTTP protokolu cez cookies nevidím ako problém, tak mal najhoršie výsledky.

Záverečná tabuľka 3.3 sa zaoberá kódovaním samotného zobrazeného dokumentu. Podstatná informácia je, či získaný obsah tvorí jeden dokument alebo séria viacerých dokumentov združených do jedného celku pomocou rámov. Možnosť získať pôvodnú a kompletnú hlavičku e-mailu je značne obmedzená. Ak server neponúka možnosť zobraziť pôvodnú správu bez zvyšku rozhrania (servery označené *), tak získať všetky pôvodné polia ani nie je možné. Server *email.sms.cz* zobrazuje niektoré názvy polí hlavičky preložené do češtiny. Pre užívateľa to môže byť výhodou, ale v tomto prípade je to skôr nežiadúce.

adresa serveru	kódovanie	rámy <frame>	polia hlavičky e-mailu
mail.google.com	utf-8	nie	všetky *
www.operamail.com	iso-8859-2, iso-8859-1	áno	From, To, CC, Subject, Date
www.centrum.cz	windows-1250	áno	všetky *
email.sms.cz	windows-1250	nie	všetky

Tabuľka 3.3: analýza HTML kódu

3.2 Analýza WWW rozhrania

Na každý server je potrebné sa prihlásiť cez HTML formulár použitím prihlasovacieho mena a hesla. Tieto údaje si volí užívateľ pri registrácii na daný server. Formulár na odoslanie údajov používa metódu POST. Môže obsahovať niekoľko skrytých polí, ktorých hodnota sa musí odoslať na adresu danú formulárom, aby bolo prihlásenie úspešné. Každý prihlasovací formulár obsahuje pole typu *password*, ale toto pole nemusí byť nutne použité ako vstup pre prihlasovacie heslo. Viac informácií je predchádzajúcej kapitole 3.1.

Po prihlásení sa zobrazí zoznam e-mailov, alebo je potrebné otvoriť jeden prípadne viac odkazov, aby sa tento zoznam zobrazil. Zoznam je organizovaný formou tabuľky. Stĺpce vždy obsahujú informáciu o predmete, odosielateľovi a dátume prijatia správy. Ich poradie nie je zaručené a môžu sa objaviť aj ďalšie stĺpce. Napríklad informácia o veľkosti e-mailu. Riadky tabuľky sú zoradené podľa dátumu zostupne z vrchu smerom nadol. Každý server povoľuje pre správy vytvárať priečinky, teda každý priečinko po otvorení obsahuje spomínaný zoznam. Predmet e-mailu je zvyčajne zároveň aj odkazom a jeho otvorením sa správa zobrazí.

Po otvorení e-mailu sa zobrazí jeho hlavička, telo a prípadné prílohy. Ďalej sa zobrazia odkazy, ktoré smerujú na ďalší a nasledujúci e-mail. Všetky e-maily je možné zobrazit' tým spôsobom, že sa otvorí prvý zo zoznamu a ďalší za ním nasledujúci. Následne sa z každého novootvoreného emailu otvorí ďalší. Pri poslednom e-maile nebude odkaz na nasledujúci existovať. Umiestnenie jednotlivých častí správy v HTML kóde nie je presne dané. Nie je možné žiadnym spôsobom jednoznačne určiť elementy, ktoré by obsahovali časti reprezentujúce e-mail. Ten sa nachádza v dokumente spolu s HTML prvkami tvoriacimi užívateľské rozhranie celého systému.

Medzi jednotlivými HTML stránkami sa udržiava stav cez cookies. To platí spravidla medzi stránkou s prihlasovacím formulárom a ostatnými časťami. Stránky môžu obsahovať rámy a v HTML kóde sa môže objaviť *meta* značka s argumentami prevádzajúcimi presmerovanie. Kódovanie stránok môže byť rozličné aj v rámci toho istého servera. Je dané buď *meta* značkou, alebo ho určuje HTTP hlavička dokumentu.

4 Návrh aplikácie

Aby mohla aplikácia pracovať automaticky tak najlepší by bol ten prístup, že jej jadro by tvorila konzolová aplikácia komunikujúca z HTTP serverom a vykonávajúca predom nastavenú činnosť. Výhodou takeého programu je možnosť bezproblémového automatického spúšťania na základe časového údajia prostriedkami, ktoré ponúka každý moderný operačný systém. Znamená to, že operačný systém by mohol tento program automaticky a periodicky spúšťať. Jadro aplikácie by len zistilo, že či pribudli nové e-maily. Jednotlivé správy spolu s príslušnými prílohami (ak existujú) by sa ukladali do súborov. Prezentácia výsledkov už môže byť nezávislá a môže mať aj grafické užívateľské rozhranie, alebo s vytvorenými súbormi môže užívateľ pracovať v ľubovoľných aplikáciách, keďže e-mail sa ukladá do obyčajného textového súboru.

Komunikačný program musí mať v sebe implementovanú aspoň časť HTTP protokolu, ktorá by mu umožňovala zaslať požiadavku typu GET alebo POST. Prvá bude využívaná pri bežnej komunikácii. Požiadavka POST sa bude využívať pri odoslaní prihlasovacích údajov na server rovnakým spôsobom ako by to bolo, keď užívateľ vyplní prihlasovacie údaje do HTML formuláru a ten odošle. Program musí byť schopný na základe stavového kódu odpovede rozhodnúť o úspešnosti alebo prípadnom zlyhaní požiadavky. Musí vedieť pracovať s cookies.

Pri ďalších vlastnostiach je kladený dôraz na výsledky analýzy obsiahnuté v kapitole 3. Aplikácia bude pracovať hlavne s dokumentom reprezentujúcim HTTP odpoveď. Prvou dôležitou úlohou je správne určiť použité kódovanie znakovkej sady, pretože to sa môže líšiť nie len pri porovnávaní jednotlivých serverov, ale aj v rámci jedného serveru. Nevýhodou je, že kódovanie odpovede môže byť určené hlavičkou alebo aj odpovedajúcou značkou v samotnom tele odpovede. Mal by existovať systém umožňujúci rovnaký prístup k obsahu bez ohľadu na použité kódovanie. Najlepšie by bolo prekódovať obsah na vnútornú reprezentáciu. Aplikácia vo svojom kódovaní môže bez problémov rozlišovať znaky s diakritikou špecifické pre väčšinu národných abecied. Obdobný prístup využívajú aj internetové vyhľadávače, ktoré dokument pred uložením prekódujú do znakovkej sady podporujúcej všetky znaky. Ak budú všetky dokumenty prezentované tým istým spôsobom, na prácu s ich obsahom sa použijú tie isté prostriedky. Okrem znakovkej sady je rovnako nejednoznačné aj určenie ďalšieho dokumentu pri obdržaní stavového kódu 3xx (tabuľka 2.2), ktorý značí presmerovanie. Adresa presmerovaného dokumentu sa môže objaviť ako v hlavičke tak aj v tele odpovede.

Jedným z tých jednoduchších prípadov, ktoré je potrebné ošetriť, je prítomnosť rámov v (X)HTML. Narozdiel od internetového prehliadača, kde definujú isté vizuálne rozloženie prvkov, v tejto aplikácii na presnom umiestení jednotlivých dokumentov v rámci jednej stránky nezáleží.

Odpoveď v tej forme ako bola obdržaná, nikdy nie je prezentovaná užívateľovi. Z toho dôvodu je úplne korektné, ak sa obsah rámov zobrazí za sebou tak, v akom poradí boli uvedené ich adresy v (X)HTML kóde dokumentu.

Posledná časť pracujúca s (X)HTML kódom je príprava POST požiadavky. Pri prístupe k pošte cez internetový prehliadač užívateľ na základe vizuálnej informácie najprv vyplní svoje prihlasovacie údaje do nejakého formuláru. Ten môže okrem vstupných polí určených pre meno a heslo, ktoré vyplní užívateľ, obsahovať aj niekoľko skrytých polí s vyplnenou hodnotou. Okrem toho dokument obsahujúci prihlasovací formulár môže obsahovať aj niekoľko ďalších formulárov s inou funkciou. Akékoľvek určovanie správneho formuláru alebo zadávanie skrytých polí by bolo obtiažne. Aplikácia preto musí mať implementovaný mechanizmus, ktorý by sám dokázal určiť, ktorý formulár slúži na prihlásenie. Ďalej musí správnym spôsobom opatriť prístupové údaje o potrebné informácie získané z tohto formuláru. Obdobným spôsobom musí získať adresu, kam treba všetky tieto údaje zaslať.

Kapitola 3.2 pojednáva o tom, že ovládanie webového rozhrania serverov a prezentácia informácií v internetovom prehliadači je vo veľkej miere podobná. Nie je však rovnaká a to je dôležitý fakt. Možno by bol zanedbateľný pre užívateľa, ktorý využíva služby poskytované webmailom. Teraz bude k tejto službe pristupovať počítačový program. Existujú dva prístupy k tomuto problému. Ten zložitejší je, že výsledná aplikácia bude natoľko sofistikovaná, že prípadné rozdiely jej nebudú robiť problém. Jednoduchšia forma riešenia je, že aplikácia bude mať pre každý server nastaviteľné správanie.

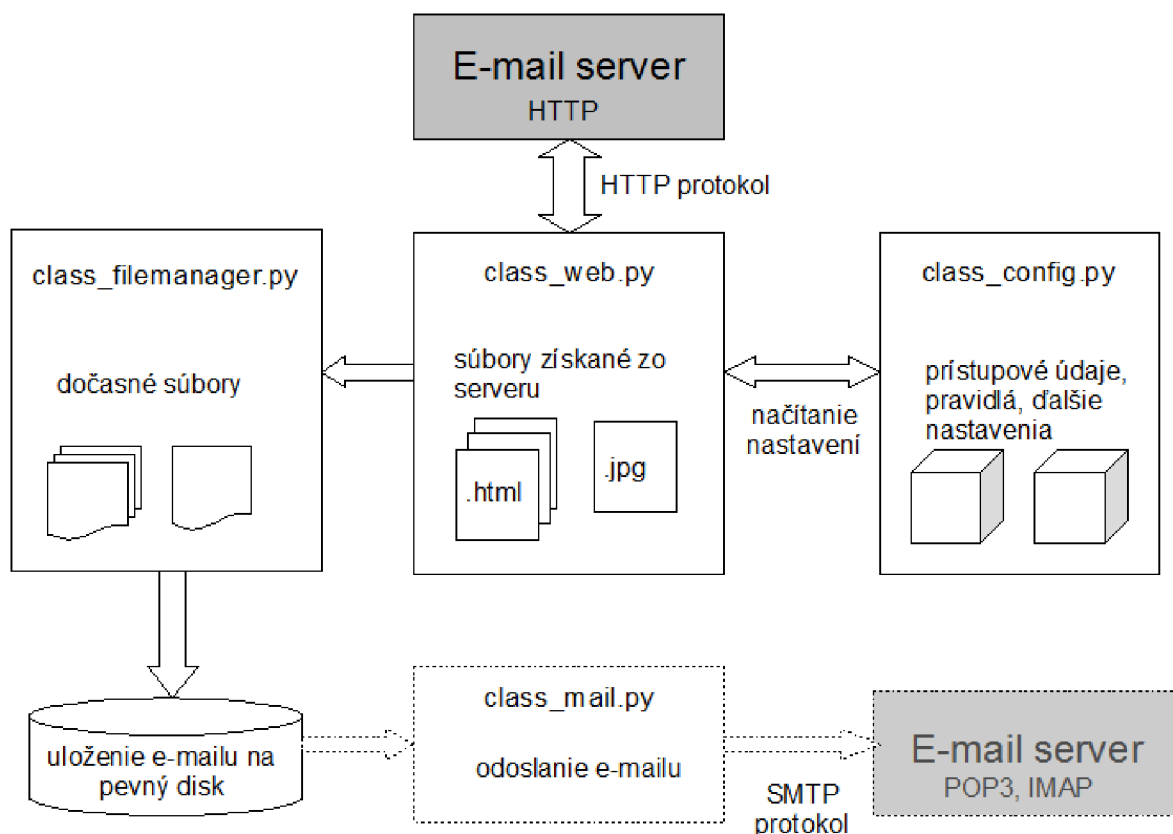
Poslednou otázkou zostáva samotné vytváranie súborov s e-mailami. Program by nikdy nemal prepísať súbor, ktorý už raz vytvoril a obdobne by nikdy nemal vytvoriť viac súborov, ktoré by reprezentovali ten istý e-mail. Týka sa to predovšetkým opakovaného prístupu k e-mailovej schránke za predpokladu, že doposiaľ uložené správy sa ešte zo servera nezmazali.

Všetky spomínané časti aplikácie by mali byť rozdelené do menších celkov.

5 Implementácia aplikácie

Ako implementačný jazyk som sa rozhodol použiť Python[14]. Bolo to hlavne z toho dôvodu, že ponúka moduly, ktoré majú v sebe implementované funkcie pre komunikáciu cez HTTP protokol. Ďalším dôležitým faktorom bola aj vhodná podpora regulárnych výrazov, ktoré sú na rôznych miestach aplikácie využívané vo veľkom množstve.

Pre lepšiu predstavu o fungovaní aplikácie je na obrázku 5.1 schéma. V ďalších podkapitolách budú jednotlivé časti detailnejšie rozpísané. Časť schémy označená bodkovanou čiarou nebola doposiaľ implementovaná. Ďalší popis neimplementovanej časti a dôvod, prečo nebol tento modul implementovaný, je v kapitole 5.1.4.



Obrázok 5.1: schéma aplikácie

Celá aplikácia bola implementovaná v znakovkej sade UTF-8⁵. Táto znakovú sadu nemá žiadny problém so špeciálnymi znakmi celého počtu národných abecied, ktoré môže obsahovať internetová stránka alebo e-mail. Znamená to, že všetky súbory, ktoré aplikácia používa, alebo ňou boli vytvorené, sú rovnako v znakovkej sade UTF-8.

5 UTF-8 – znaková sada obsahujúca znaky všetkých svetovo používaných jazykov

Jadro aplikácie je tvorené programom, a ten využíva služby všetkých modulov. Pred zahájením komunikácie so serverom sa najprv načítajú nastavenia pomocou služieb konfiguračného modulu. Ten je potrebný iba pri štarte programu.

5.1 Modularita

Aplikáciu tvoria tri samostatné moduly. Prvý modul slúži na konfiguráciu a uloženie nastavení aplikácie. Ďalší zabezpečuje komunikáciu cez HTTP protokol a posledný má na starosti ukladanie súborov na disk. Zvolil som v nich objektový prístup, takže sa v každom nachádza najmenej jedna trieda, ktorá implementuje potrebné metódy. Mená jednotlivých modulov sú uvedené na obrázku 5.1.

5.1.1 Modul pre nastavenie aplikácie

Tento modul slúži na zmenu nastavení, ich zápis do súboru a prípadné ich načítanie zo súboru pri ďalšom spustení aplikácie. Aplikácie používa hlavný a užívateľom definovaný konfiguračný súbor. Konfiguračný súbor pozostáva zo sekcií. Názov sekcie je uzatvorený v zátvorkách []. Za názvom nasleduje telo sekcie, ktoré nie je povinné. Telo obsahuje záznamy, pozostávajúce s názvu premennej a hodnoty premennej. Jednotlivé záznamy, ale aj názov sekcie sú navzájom oddelené aspoň jedným prázdny riadkom. S takouto štruktúrou súboru dokáže vhodne pracovať štandardná trieda ConfigParser [12], pomocou ktorej bol tento modul implementovaný.

Hlavný konfiguračný súbor obsahuje predovšetkým názov užívateľského konfiguračného súboru. Tento prístup umožňuje vytvoriť niekoľko súborov s požadovanými nastaveniami a potom už len zmeniť názov, aby sa nastavenia začali používať.

Užívateľom definovaný konfiguračný súbor neobsahuje len prístupové údaje, ale aj polia HTTP hlavičky, ktoré odosiela aplikácia na server. Okrem týchto údajov súbor obsahuje aj kompletný popis správania programu alebo akcií, ktoré majú byť po pripojení sa k serveru vykonané. Akcie definujú postup získania jednotlivých e-mailov z (X)HTML kódu. V popise správania programu sú uvedené adresy dokumentov, ktoré má program otvoriť. Pritom je možné získať adresu ďalšieho dokumentu z predchádzajúceho na základe odkazu. To znamená, že aspoň prvý dokument musí byť zadaný jednoznačne. Adresa ďalšieho môže byť názov odkazu, ktorý obsahuje nejaký predchádzajúci dokument. Ak aplikácia otvorí a získa dokument textového formátu (sem patrí aj (X)HTML dokument), tak z neho môže uložiť iba nejakú časť a zvyšok dokumentu bude vypustený. Získaná časť dokumentu bude reprezentovať užitočnú informáciu akou je napríklad telo e-mailovej správy, polia jej hlavičky alebo príslušná príloha. Uvedené položky e-mailovej správy sú uvedené v nastaveniach ako regulárne výrazy. Keďže každý server používa vlastné užívateľské rozhranie, tak aj regulárne výrazy sú pre každý server odlišné. Nakoniec je v nastaveniach akcií aplikácie uvedené, že

ktoré dokumenty alebo časti dokumentov je potrebné uložiť na disk, aby vo výslednom súbore bola kompletná informácia o jednej e-mailovej správe. Vhodným spôsobom je vyriešené, aby sa príslušné akcie vykonali nad všetkými e-mailami.

Implementovaný modul dokáže okrem zápisu a čítania nastavení odhaliť niektoré syntaktické chyby, ktoré mohli vzniknúť pokusom o zápis nesprávnej hodnoty cez tento modul, alebo manuálnym editovaním nastavení.

5.1.2 Komunikačný modul

Jadro aplikácie tvorí komunikácia cez HTTP protokol. Komunikačný modul bol navrhnutý za týmto účelom. Zoskupuje metódy, ktoré dokážu komunikovať s HTTP serverom a čítať jeho odpovede, ale aj metódy vykonávajúce analýzu obdržaného HTML kódu.

Jednou zo základných funkcií je získanie objektu na základe URL adresy. Každý objekt poslaný od serveru je modul schopný uchovať vo vlastnej réžii bez nutnosti uloženia do súboru. Znamená to, že modul uchováva získané objekty v asociatívnom poli uloženom v operačnej pamäti a v prípade potreby môže k nim pristupovať. Neukladá si každý jeden objekt, pretože by to bolo náročné na pamäťové nároky, ale len tie, ktoré budú neskôr potrebné. O objekte eviduje okrem jeho vlastného obsahu aj adresu, z ktorej bol získaný a jeho typ (obrázok, video atď.). Uvedený prístup je výhodný, hlavne ak sa jedná o (X)HTML dokument. Takýto dokument môže obsahovať odkazy na ďalšie dokumenty a objekty. Aplikácia je potom schopná získať ďalší objekt prostredníctvom existujúceho odkazu. Adresa odkazovaného objektu môže byť zadaná absolútne alebo relatívne vzhľadom k adresárovej štruktúre serveru, prípadne môže smerovať na iný server. Modul dokáže rozhodnúť o aký typ adresy sa jedná a zvolí správny postup pri jej otvorení. Je možné žiadať o získanie objektu podľa názvu odkazu alebo adresy odkazu. Obe hodnoty pritom môžu byť zadané regulárnym výrazom.

Pri obdržaní stavového kódu značiaceho presmerovanie sa modul snaží o načítanie novej adresy objektu. Novú adresu je modul schopný získať aj z tela HTTP odpovede. Každý získaný objekt si modul prekóduje do znakovkej sady UTF-8, aby vedel s daným obsahom pracovať. Tento postup bol zvolený hlavne kvôli znakom národných abecied, ktoré by inak boli nečitateľné.

Poslednou funkciou modulu je prihlásenie užívateľa cez webový formulár. Pri použití tejto akcie sa najprv vyhľadá vhodný formulár v obsahu dokumentu slúžiaceho na prihlásenie. Následne je nájdený formulár analyzovaný a je vyhľadaná adresa, kam sa majú jeho údaje odoslať. Nakoniec sa pripraví skupina hodnôt, ktoré sú odoslané na požadovanú adresu.

Pri komunikácii so serverom modul uchováva cookies (kapitola 3.1), ktoré získava z HTTP hlavičky odpovede. Pri komunikácii ich hodnoty posiela serveru. Pre uloženie cookies bola zvolená jednoduchšia forma. Neberie sa v úvahu časová doba, po ktorú je hodnota cookies platná. Všetky

hodnoty sú posielané na server od momentu ich nastavenia po skončenie aplikácie. Tým pádom sa počas behu programu nedajú odstrániť. Rovnako sa neprihliada na platnosť v rámci adresárovej štruktúry serveru. Hodnoty sú prístupné globálne. V niektorých prípadoch môže nastať situácia, že komunikácia počas jedného behu prechádza cez niekoľko serverov (prihlasovací server nie je ten istý ako server, kde je uložená pošta). V tomto prípade sú hodnoty poslané na oba servery. Tento prístup sa osvedčil ako dostačujúci a nepriniesol žiadne problémy.

5.1.3 Modul pre správu súborov

Správy sú na e-mailovom serveri jednoznačne rozlíšiteľné na základe identifikátora. Pri ukladaní na disk aplikácia zväčša disponuje len informáciou o tom, aký má správa predmet. Správa sa uloží na disk pod rovnakým menom ako je jej predmet. Prípadne je názov súboru vždy ten istý, ale súbor je umiestnený v adresári s totožným názvom ako je predmet. Pri tomto prístupe môžu nastať dve nežiadúce situácie a modul bol navrhnutý na ich riešenie.

Prvou z nich je nevhodná dĺžka predmetu správy. Správa bez predmetu je vzhľadom na formát e-mailu úplne v poriadku. Aj túto správu je potrebné nejakým spôsobom pomenovať ako súbor na pevnom disku. Modul ošetruje pokus o zápis do súboru bez názvu. Opačný ale rovnako nežiadúci je stav, že predmet je neprimerane dlhý. Aj keď konkrétna dĺžka súboru je spätá s operačným systémom a v dnešnej dobe by nemal byť problém vytvárať súbory aj s veľmi dlhými názvami, tak akýkoľvek pokus o vytvorenie neprimerane dlhého názvu súboru alebo adresára je v tichosti ignorovaný a názov je skrátený na vhodný počet znakov.

Ďalšia situácia sa týka znakovkej sady použitej v názve súboru. Ten nemôže obsahovať ľubovoľné znaky. Je to závislé od operačného systému. Aby som sa napriek tomu vyhol zbytočným komplikáciám, tak názov súboru je pred vytvorením upravený, aby obsahoval len znaky anglickej abecedy, čísla, znak bodka (.) a znak podtržník (_). Vytvoriť názov súboru s diakritikou bolo považované tiež za problematické z dôvodu prenositeľnosti. Aplikácia by musela mať informáciu o používanom súborovom systéme, a preto bolo z tejto možnosti upustené.

Veľmi jednoducho by mohla nastať situácia, že dve správy majú rovnaký predmet. Aby nedošlo k prepísaniu obsahu súboru tak modul je schopný vhodne upraviť meno súboru na jedinečnú hodnotu. Túto úpravu vykoná len ak je obsah súborov rôzny. Aj keď Python disponuje modulom, ktorý dokáže rozhodnúť o zhode súborov, použitá metóda sa mi nejavila ako postačujúca. Porovnanie bolo implementovaná tak, že najskôr sa porovná veľkosť súborov. Ak je rovnaká, z oboch súborov sa vypočíta md5 hash[13] a tieto hodnoty sa porovnávajú. Ich zhoda znamená, že aj obsah súborov je rovnaký a nedochádza ku žiadnemu zápisu do súboru. Môže sa to zdať komplikované, ale pri väčších súboroch je takéto porovnanie rýchlejšie.

5.1.4 Modul pre odosielanie e-mailov (neimplementované)

V návrhu aplikácie bola zmienka o tom, že e-mailové správy sa ukladajú do súborov a ich prezentácia môže byť ľubovoľná. Jedným z vhodných riešení, by bolo odosielanie uložených e-mailov na iný e-mailový server, na ktorý by mal užívateľ lepší prístup cez IMAP protokol. Takto by bolo možné združovať správy aj z veľkého počtu voľne dostupných e-mailových serverov na jednom serveri, kde by boli uložené. Prístup k nim by bol cez existujúceho poštového klienta.

E-mailové správy na webovom rozhraní nemajú formát popísaný v kapitole 2.1. Takže tento formát by sa musel pre odoslaním rekonštruovať. Táto akcia by vôbec nebola zložitá, ale hlavným problémom je, že e-maily sa ukladajú na disk v takom formáte, ako to bolo uvedené v nastaveniach. To znamená, že ak to v nastaveniach nie je uvedené, tak uložená správa nemusí obsahovať vôbec žiadnu hlavičku. Okrem toho sa e-maily môžu ukladať aj v rámci zložitejšej adresárovej štruktúry .

Modul odosielaajúci správy na IMAP server by bol závislý od spôsobu ukladanie správ na pevnom disku. Ak by sa tento spôsob v nastaveniach zmenil, tak by sa musel zmeniť aj samotný zdrojový kód tohto modulu. Z toho dôvodu by implementácia modulu, ktorý by posielal uložené správy zatiaľ nemala zmysel.

5.2 Výsledná aplikácia

Konečná aplikácia má podobu dvoch programov. Prvý z nich slúži iba na zápis nastavení aplikácie (používa iba modul 5.1.1) a z hľadiska celkovej práce nie je ďalej podstatný. Druhá aplikácia používa všetky vyššie popísané moduly. Najprv prečíta nastavenia aplikácie. Pred pripojením sa k serveru analyzuje pravidlá. Ak sú pravidlá syntakticky správne vytvorí si ich vnútornú reprezentáciu. Na základe pravidiel riadi ďalšiu činnosť, pri ktorej volá metódy implementované vo zvyšných moduloch (5.1.2 a 5.1.3).

Pri implementácii bolo myslené na jednoduchý spôsob úpravy modulov alebo aj celej aplikácie. Z toho dôvodu bola aplikácie rozdelená do viacerých funkčných modulov a dvoch programov, ktoré sa starajú o nastavenia a samotný beh. Takto implementované jadro aplikácie tvorí iba jednoduchý interpret pravidiel. Pri vylepšení aplikácie je bez problémov možné používať funkčnosť modulov, tým spôsobom, že sa prerobí iba hlavné jadro programu. Ďalšiemu vývoju sa venuje posledná kapitola 7.1.

6 Výsledky

Aplikácia bola implementovaná tak, aby spĺňala nároky kladené v predchádzajúcich kapitolách. Bol vyvinutý konfiguračný súbor, vďaka ktorému je možné uložiť na disk e-mailové správy zo serverov spomínaných v kapitole 3. Vytvorené súbory sú vhodné na ďalšie spracovanie v iných aplikáciách.

Môže sa stať, že beh aplikácie skončí chybovou správou, ktorú vypíše interpret jazyka Python. Tieto chyby sú s najväčšou pravdepodobnosťou spôsobené vnútornými záležitosťami použitých štandardných modulov. Neznačená to, že by boli nesprávne implementované. Následkom chybovej hlásky je neošetrená výnimka, vzniknutá volaním štandardnej funkcie, ktorá nie je dokumentovaná pri popise funkcie. Jej výskyt sa tým pádom nedá nijak predpokladať.

Funkčnosť programu bola testovaná na operačnom systéme Microsoft Windows (architektúra 32 bitov aj 64 bitov) a Linux s jadrom 2.6.

www.operamail.com – aplikácia vytvorí pre každý e-mail samostatný adresár, kam uloží hlavičku a telo správy. Hlavička obsahuje okrem povinných polí aj predmet. Z celej správy sú odstránené prvky značkovacieho jazyka použitého pri tvorbe WWW rozhrania serveru. V prípade, že správa obsahovala aj prílohy, tak sú všetky uložené do jednotlivých súborov v rovnakom formáte ako boli priložené.

email.sms.cz – miernou modifikáciou nastavení aplikácie bol dosiahnutý totožný výsledok ako pri predchádzajúcom serveri.

www.centrum.cz – nastavenia pre tento server sú ešte jednoduchšie, pretože ponúka možnosť zobraziť správu v rovnakom formáte s akým pracuje štandardný e-mailový klient. Správa so všetkými prípadnými prílohami je uložená do jedného súboru.

mail.google.com – server umožňuje bezplatný prístup k pošte cez POP3 aj IMAP protokol, preto by ukladania mailov cez WWW rozhranie nemalo význam. Aplikácia by nemala mať problém ani s týmto serverom.

www.o2active.cz – už pri samotnom návrhu aplikácie bolo zrejmé, že z tohto serveru ukladať e-mailové správy nebude možné. Pre zobrazenie e-mailu v internetovom prehliadači je potrebná podpora jazyka JavaScript. Týmto jazykom nedisponuje ani aplikácia, preto nedokáže uložiť e-maily z tohto serveru a ani zo žiadneho iného, kde by bol obsah WWW stránok tvorený dynamicky na strane klienta.

7 Záver

Celkový výsledok práce je možné považovať za úspešný. Vytvoreným programom bolo demonštrované, že automatický prístup k e-mailovej pošte cez WWW rozhranie je možný. Program bol otestovaný na štyroch nezávislých serveroch. Na troch z nich bol úspešný, ale pri poslednom zlyhal, pretože daný server používal prostriedok na generovanie obsahu na strane klienta.

Niet pochýb, že vytvorená aplikácia je z užívateľského hľadiska nevhodná, pretože jej správne nastavenie vyžaduje programátorské znalosti. Žiadna podobná aplikácia doteraz vytvorená nebola. Hlavný prínos tejto práce je v tom, že bol otestovaný automatický prístup k službe, ktorá nikdy nebola pre automatický prístup navrhnutá.

Okrem primárneho účelu, automatického sťahovania e-mailov cez WWW rozhranie, môže byť aplikácia použitá aj pri odlišných činnostiach. Vhodnými nastaveniami by bolo možné napríklad automaticky ukladať príspevky z internetovej diskusie. Primárne zameranie je však vhodné pre prístup k informáciám, ktoré vyžadujú autorizáciu na základe webového formuláru. V opačnom prípade by bolo vhodnejšie využiť jednoduchšie nástroje vyvinuté pre jednoduchšie účely.

Webové služby ale aj elektronická komunikácia prešla istým vývojom a z dôvodu rozsahu práce som sa nemohol venovať nastolenej problematike do väčších detailov. Pokúsil som sa stručnou formou vybrať pre čitateľa tie základné informácie, ktoré môžem v ďalšej práci rozšíriť.

7.1 Ďalší vývoj aplikácie

V ďalšom rozvoji aplikácie, ale aj celej práce, by som chcel pokračovať, pretože sa jedná o veľmi zaujímavú tému. Implementovaná aplikácia má zatiaľ ťažkopádny systém nastavení. Jedno z možných rozšírení by tým pádom mohla byť grafická nadstavba pre konfiguráciu aplikácie. Pre užívateľa by bola vhodná aj grafická prezentácia výsledkov.

Nevýhodou je aj činnosť aplikácie riadená pomocou konfiguračného súboru. V nasledujúcej verzii by som chcel do programu implementovať prvky umelej inteligencie, ktoré by aplikácii umožňovali rozhodovať sa. Väčšina informácií na e-mailových serveroch je stále v textovej forme. Aplikácia by sama bola schopná na základe získaného textu otvoriť dokument prezentujúci e-mailovú správu, v ktorej by našla hlavičku a telo. Pre tento prístup by bola nutná analýza oveľa väčšieho počtu webových rozhraní. Aplikácia by sa rozhodovala na základe kľúčových slov.

Odstránenie časti nastavení by prinieslo aj ďalšiu výhodu. Teraz je formát uloženého e-mailu na pevnom disku závislý od toho, aké sú nastavenia. Prílohy sa ukládajú do samostatných súborov. Ak by aplikácia dokázala automaticky rozpoznať v (X)HTML kóde jednotlivé časti e-mailovej správy (hlavička, telo a prílohy), tak formát uloženia by určoval program a nebol by modifikovateľný.

Najlepší prístup by bol vytvárať rovnaký formát uložených správ zhodný s MIME formátom. To znamená, že celá správa vrátane príloh by bola uložená v jednom súbore. Tvorba takého formátu si však vyžaduje ďalšie štúdium. Po implementovaní tohto rozširujúceho prvku, bude mať zmysel implementovať aj modul (kapitola 5.1.4), ktorý by uložené e-maily posielal na IMAP server.

Iným možným rozšírením je implementácia nejakého druhu interpreta jazyka JavaScript. JavaScript, ktorý obsahujú webové stránky, je označovaný ako *klientský JavaScript*. Pre správnu podporu jazyka by stačilo, aby aplikácia mala implementovanú formu označovanú *Core JavaScript*.

Literatúra

- [1] BOS, Bert. *Cascading Style Sheets* [online]. 2008 [cit. 2008-04-01].
Dostupný z WWW: <<http://www.w3.org/Style/CSS/>>.
- [2] CRISPIN, Mark R.. *RFC 3501 : INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1* [online]. © 2003 [cit. 2008-04-02].
Dostupný z WWW: <<http://www.faqs.org/rfcs/rfc3501.html>>.
- [3] FIELDING, R., et al. *RFC 2616 : Hypertext Transfer Protocol - HTTP/1.1* [online].
The Internet Society, © 1999 [cit. 2008-03-26].
Dostupný z WWW: <<http://www.faqs.org/rfcs/rfc2616.html>>.
- [4] FIELDING, R., et al. *RFC 2616 : Hypertext Transfer Protocol - HTTP/1.1* [online].
The Internet Society, © 1999 [cit. 2008-03-26]. p 27
Dostupný z WWW: <<http://www.faqs.org/rfcs/rfc2616.html>>.
- [5] FREED, Ned, BORENSTEIN, Nathaniel S.. *RFC 2045 : Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies* [online].
[1996] [cit. 2008-03-16]. Dostupný z WWW: <<http://www.faqs.org/rfcs/rfc2045.html>>.
- [6] FREED, Ned, BORENSTEIN, Nathaniel S.. *RFC 2046 : Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types* [online]. 1996 [cit. 2008-03-16].
Dostupný z WWW: <<http://www.faqs.org/rfcs/rfc2046.html>>.
- [7] INTERNET SOFTWARE CONSORTIUM. *Lynx* [online]. [2008] [cit. 2008-03-02].
Dostupný z WWW: <<http://lynx.isc.org/>>.
- [8] KRISTOL, David M., MONTULLI, Lou. *RFC 2109 : HTTP State Management Mechanism* [online]. [1997] [cit. 2008-04-06].
Dostupný z WWW: <<http://www.faqs.org/rfcs/rfc2109.html>>.
- [9] MOORE, Keith. *RFC 2047 : MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text* [online]. [1996] [cit. 2008-03-16].
Dostupný z WWW: <<http://www.faqs.org/rfcs/rfc2047.html>>.
- [10] MOZILLA FOUNDATION. *JavaScript* [online]. [2008] [cit. 2008-03-05].
Dostupný z WWW: <<http://developer.mozilla.org/en/docs/JavaScript>>.
- [11] MYERS, John G., ROSE, Marshall T.. *RFC 1939 : Post Office Protocol - Version 3* [online].
[1996] [cit. 2008-04-02]. Dostupný z WWW: <<http://www.faqs.org/rfcs/rfc1939.html>>.
- [12] PYTHON SOFTWARE FOUNDATION. *Filecmp : File and Directory Comparisons* [online].
2008 [cit. 2008-04-06]. Dostupný z WWW: <<http://docs.python.org/lib/module-filecmp.html>>.
- [13] PYTHON SOFTWARE FOUNDATION. *MD5 : MD5 message digest algorithm* [online].
2008 [cit. 2008-04-06]. Dostupný z WWW: <<http://docs.python.org/lib/module-md5.html>>.

- [14] PYTHON SOFTWARE FOUNDATION. *Python Programming Language* [online]. © 1990-2008 [cit. 2008-04-10]. Dostupný z WWW: <<http://www.python.org>>.
- [15] RESNICK, P.. *RFC 2822 : Internet Message Format* [online]. The Internet Society, © 2001 [cit. 2008-03-15]. Dostupný z WWW: <<http://www.faqs.org/rfcs/rfc2822.html>>.
- [16] W3C. *Markup Validation Service* [online]. © 1994-2007 [cit. 2008-03-05]. Dostupný z WWW: <<http://validator.w3.org/>>.
- [17] W3C. *XHTML2 Working Group Home Page* [online]. © 1995-2007 [cit. 2008-03-05]. Dostupný z WWW: <<http://www.w3.org/MarkUp/>>.

Zoznam príloh

Príloha A CD-ROM

A CD-ROM

K práci je priložené CD, ktoré obsahuje:

- Text tejto práce vo formáte PDF, aby ju bolo možné znovu vytlačiť,
- zdrojové texty vytvorenej aplikácie,
- užívateľskú príručku k aplikácii ako samostatný dokument vo formáte PDF.