Elektronická pošta

Miroslav Týnovský

17. srpna 2003

Obsah

1 Úvod

Elektronická pošta je jednou ze základních a snad nejčastěji používanou internetovou službou. Používat ji umí leckdo, stačí si spustit klienta, napsat zprávu a odeslat. Jak se však zpráva dostane po síti až k příjemci? To by měl popsat a vysvětlit následující text. Dále se bude zabývat formátem zprávy a poštovními protokoly.

2 Architektura elektronické pošty

Jak se zpráva dostane k adresátovi? Na počátku je program, který odesílatel používá jako svého e-mailového klienta, tzv. *MUA – Mail User Agent*. Mezi představitele těchto programů patří například *Pine*, *Mutt*, *Paegasus Mail* nebo *Microsoft Outlook*. Úkolem MUA je samozřejmě poskytnout uživateli rozhraní pro napsání zprávy, ale hlavně zprávu odeslat. Jeho "odeslání" je však jenom uložení do fronty (lokální nebo vzdálené), ze které si ji vyzvedne program MTA.

MTA – Mail Transfer Agent převezme zprávu od MUA, a potom podle doménové části adresy příjemce musí určit server, který je co nejblíže k adresátovi. Pak otevře spojení a pokusí se zprávu předat MTA na tomto serveru pomocí protokolu *SMTP*. Takto je zpráva mezi servery předávána, až dorazí na server příjemce. Pokud se sem zprávu nepodaří doručit, měl by poslední dostupný MTA určit, zda se jedná o dočasný problém, v tom případě pokus o odeslání opakovat později, nebo o problém neodstranitelný, v tom případě ohlásit chybu. Posledním v řetězci je tedy přijímající MTA na serveru příjemce. Ten zkontroluje jmennou část adresy a uloží zprávu do schránky s tímto jménem. Reprezentanty MTA jsou například programy *sendmail*, *qmail*, *Exim* a *Postfix*.

Přijímající MTA pro doručení do schránky často používá tzv. *LDA – Local Delivery Agent*. Ten na zprávu ještě může aplikovat filtry, třeba uložení do různých schránek, automatická odpověď, přeposlání apod. Mezi nejznámější LDA patří *procmail* a *maildrop*.

Na tomto místě se nám doručovací mechanizmus rozpadá na dvě varianty. Pokud si adresát vybírá poštu pomocí MUA, který je přímo na serveru, je vlastně vše hotovo. Stačí jej spustit a číst. Pokud si adresát vybírá poštu pomocí MUA na svém osobním počítači, musí tento MUA ještě zprávy ze serveru stáhnout. K tomu slouží protokoly *POP* a *IMAP*.

Celý postup doručení zprávy je znázorněn na Obrázku 1.

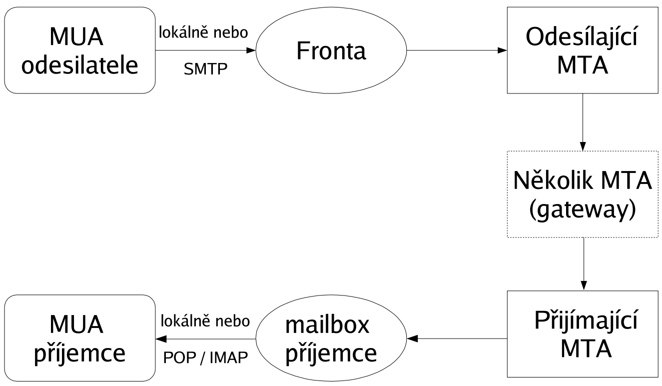
3 Formát zprávy

3.1 Hlavičky a tělo

Když píšeme zprávu, zadáme obvykle předmět (`Subject`), adresu příjemce (`To`), případně příjemce kopie dopisu (`Cc`) nebo příjemce tzv. slepé kopie (`Bcc`). MUA doplní datum a čas odeslání (`Date`), adresu odesílatele (`From`), jednoznačnou identifikaci zprávy (`Message-Id`), případně i další údaje a z každé této informace vytvoří tzv. hlavičku. Samotný text dopisu pak tvoří tělo a je od hlaviček odděleno prázdnou řádkou. Během doručování dopisu mohou MTA přidávat ještě další hlavičky (například `Received`).

V hlavičkách mají následující znaky zvláštní význam:

- středník `;` a dvojtečka `:` mají význam oddělovače v seznamu (například při oddělování adresátů v hlavičce `To`).



Obrázek 1: Architektura elektronické pošty

- špičaté závorky `<>` mají speciální význam v adrese; vše mimo řetězec ve špičatých závorkách se ignoruje.
- hranaté závorky `[]` mají význam ve jméně počítače; sdělují, že jméno počítače nemá být překládáno v DNS.
- do kulatých závorek `()` se uzavírá komentář.

Každá hlavička začíná na novém řádku a má tvar `Název_položky: Hodnota`. Hlavička může být rozdělena na více řádků, každý další řádek pak musí začínat alespoň jednou mezerou nebo tabulátorem. Údaje v hlavičkách slouží příjemci dopisu, dozví se z nich kdo a kdy dopis odeslal, pro koho je určen, jakými servery prošel apod.

3.2 Přehled všech základních hlaviček

`Received`
: Tuto hlavičku připisuje na začátek zprávy každý e-mailový server, přes který zpráva projde. V této hlavičce se mohou vyskytovat slova:

  - `from` – počítač, ze kterého byla zpráva přijata
  - `by` – počítač, kterým byla zpráva přijata
  - `via` – fyzická cesta
  - `with` – síťový nebo poštovní protokol
  - `id` – příjemcova identifikace zprávy
  - `for` – pro koho je zpráva určena

`From`
: Od (Odesilatel).

`Sender`	Kdo vyřizuje odeslání (například konference, přes kterou zpráva přišla).
`Date`	Datum odeslání.
`Reply-To`	Adresa pro odpovědi.
`In-Reply-To`	Identifikace zprávy, na kterou tato zpráva odpovídá.
`To`	Komu (Adresát).
`Cc`	Kopie.
`Bcc`	Tajná kopie – tato hlavička se před zasláním smaže.
`Message-Id`	Identifikace zprávy.
`Keywords`	Klíčová slova.
`References`	Odkazy.
`Subject`	Předmět.
`Comments`	Komentář.
`Encrypted`	Šifrováno.
`X-...`	Uživatelsky definovaná hlavička.
`Resent-...`	Předpona před původními hlavičkami při automatickém přeposlání zprávy.

3.3 Příklad zprávy

```
Return-Path: <odesilatel@informatics.muni.cz>
Delivered-To: alias-email-adresat@seznam.cz
Received: (qmail 1114 invoked from network); 25 Jul 2003 16:29:43 -0000
Received: from unknown (HELO anor.ics.muni.cz) (147.251.4.35)
  by hermes.go.seznam.cz with SMTP; 25 Jul 2003 16:29:43 -0000
Received: from anxur.fi.muni.cz (IDENT:0@anxur.fi.muni.cz [147.251.48.3])
  by anor.ics.muni.cz (8.12.1/8.12.1) with ESMTP id h6PGTgms005104;
  Fri, 25 Jul 2003 18:29:42 +0200
Received: from aisa.fi.muni.cz (IDENT:0@aisa [147.251.48.1])
  by anxur.fi.muni.cz (8.12.9/8.12.8) with ESMTP id h6PGTgPN012717;
  Fri, 25 Jul 2003 18:29:42 +0200 (MEST)
Received: from aisa.fi.muni.cz (IDENT:14238@localhost [127.0.0.1])
  by aisa.fi.muni.cz (8.12.9/8.12.8) with ESMTP id h6PGTfED822479;
  Fri, 25 Jul 2003 18:29:41 +0200 (MET DST)
Received: from localhost (odesilatel@localhost)
  by aisa.fi.muni.cz (8.12.9/8.12.5/Submit) with ESMTP id h6PGTfaa774634;
  Fri, 25 Jul 2003 18:29:41 +0200 (MET DST)
Date: Fri, 25 Jul 2003 18:29:40 +0200
From: Odesilatel <odesilatel@informatics.muni.cz>
To: adresat@seznam.cz,
Subject: Dlouho ocekavana oslava (fwd)
Message-Id: <Pine.SGI.4.21.0307251821370.783105-100000@aisa.fi.muni.cz>
Mime-Version: 1.0
Content-Type: TEXT/PLAIN;
  charset=US-ASCII
X-Muni-Spam-Testip: 147.251.48.3
```

```
X-Muni-Envelope-From: odesilatel@informatics.muni.cz
X-Muni-Virus-Test: Clean
Status: R
X-Status: N
X-KMail-EncryptionState:
X-KMail-SignatureState:

Text zprávy
```

3.4 Obálka

Pro doručování dopisu mezi MTA však nejsou informace v hlavičce směrodatné! MTA se řídí informacemi uloženými v tzv. *obálce (envelope)*, kterou vytvoří první MTA, který dopis dostane od MUA. Do obálky je zařazena adresa odesílatele a jedna nebo více adres příjemců, kterou sdělí MUA prvnímu MTA. Zde je vysvětlení, proč dojde zpráva i adresám uvedeným v `Bcc`, jsou totiž zahrnuty do obálky a ne do hlavičky. Tím pádem o nich ostatní příjemci nebudou vědět (narozdíl od těch, kteří jsou uvedeni v `Cc`).

Důvodů, proč jsou adresy v hlavičkách i v obálce je více:

- dopisy pro více příjemců cestují jen v jednom exempláři až do místa, kde se jednotlivé kopie rozcházejí – pak MTA z obálek jednotlivých kopií odstraní příjemce, pro něž je dopis odesílán jinou cestou. Všichni příjemci dostanou identické kopie dopisu a MTA dopis podle položky `To` nerozmnoží.
- uživatel si může nechat dopis forwardovat na jiný účet a přijde v nezměněné formě. MTA totiž přepíše adresu příjemce pouze v obálce, v hlavičce zůstane ta původní.
- je-li zpráva nedoručitelná, vrátí ji MTA zpět, jako odesílatel se uvede pouze do hlavičky (`From`) a ne do obálky, tudíž předejde zbytečné chybové zprávě, kdyby i tato zpráva původnímu odesílateli byla nedoručitelná.
- při rozesílání dopisů do konference je adresa odesílatele v obálce nahrazena adresou správce konference, který pak bude přijímat chybové zprávy a může tak sledovat úspěšnost doručování zpráv v konferenci a nefunkční adresy vymazat ze seznamu.
- dopisy z konferencí mají většinou v hlavičce `To` uvedeno jen "Příjemci konference XXX" a adresa příjemce je uvedena jen v obálce. Umožňuje to lepší orientaci v dopisech. Též v položce `From` je uveden skutečný odesílatel místo automatu, který dopisy skutečně množí a distribuuje mezi členy konference.

4 MIME

Formát zpráv původně umožňoval přenášet data pouze ve kódování ASCII (7 bitů). Brzy se ale ukázalo, že tento standard nedostačuje pro požadavky uživatelů, kteří chtějí posílat i texty jiných znakových sad, formátovaný text, obrázky, multimediální soubory a jiné binární soubory. Z toho důvodu bylo uvedeno a standardizováno *MIME – Multipurpose Internet Mail Extension*, rozšíření, které tyto služby umožňuje.

Zaslat zprávu obsahující jiná data než text v ASCII lze i bez MIME. Je však nutné před odesláním zprávu zakódovat do ASCII. Příjemce po výběru zprávu zase dekóduje. Používaná kódování jsou například *Base 64*, *Quoted Printable* a *UUENCODE*. Zkušený příjemce podle formátu dat pozná, jaký dekódovací program použít.

Tento postup se dá pomocí zautomatizovat pomocí dodatečných údajů v hlavičkách, popisujících typ přenášených dat a kódovací algoritmus. Formát těchto hlaviček specifikuje právě norma MIME.

4.1 MIME hlavičky

MIME zavádí hlavičky:

- `MIME-Version` – specifiuje použitou verzi MIME (současná verze je 1.0 popsaná normami RFC-2045 až 2049)
- `Content-Type` – specifikuje typ a podtyp dat posílaných v těle zprávy
- `Content-Transfer-Encoding` – specifikuje kódování do ASCII použité na přenos
- `Content-ID` – identifikace zprávy
- `Content-Description` – textový popis obsahu
- `Content-Disposition` – určuje, zda mají být data zobrazena přímo, nebo jako příloha
- `Content-`*řetězec* je rezervováno pro budoucí použití

4.1.1 Hlavička MIME-Version

Tato hlavička specifikuje verzi MIME a je používána hlavně kvůli zajištění kompatibility. Říká klientovi, podle jaké verze byla zpráva rozšířena, ale také to, že vůbec jde o zprávu rozšířenou podle MIME. Hlavička musí být uvedena před ostatními hlavičkami MIME a je tvaru
`MIME-Version: 1.0`

4.1.2 Hlavička Content-Type

Popisuje typ dat obsažených v těle zprávy tzv. media typ, aby klient mohl zvolit vhodný způsob reprezentace obsahu zprávy. Hlavička je tvaru
`Content-Type: typ/podtyp; parametry`, kde parametry mají tvar `; atribut=hodnota`.

Typy dat mohou být:

- *složené* – to znamená, že zpráva se skládá z několika částí. Jsou to typy message, multipart nebo report. V případě složeného typu je v hlavičce `Content-Type` uveden parametr `boundary` (hranice) definující řetězec použitý v oddělovačích jednotlivých částí, které mají tvar `CRLF--boundaryCRLF`.
- *jednoduché* – některého z typů text, image, audio, video nebo application. Vždy musí být uveden podtyp a volitelně mohou následovat parametry, např. český text s diakritikou v kódování ISO-8859-2 by měl být označen
`Content-Type: text/plain; charset=iso-8859-2`

4.1.3 Hlavička Content-Transfer-Encoding

Určuje, jak jsou data pro přenos zakódována. MIME definuje dva algoritmy kódování dat: Quoted-Printable a Base64. `Content-Transfer-Encoding` tak nejčastěji nese následující způsoby kódování:

- `Quoted-Printable`
- `Base64`
- `7bit`[1] – data nejsou kódována, obsahují pouze znaky ASCII, řádky obsahují maximálně 998 znaků (bez `CRLF`)
- `8bit` – data nejsou kódována, řádky obsahují maximálně 998 znaků
- `binary` – data nejsou kódována, jsou libovolná
- `x-`... – experimentální kódování

Tato hlavička se vztahuje k celému tělu zprávy, ale může se objevit v nějaké konkrétní části zprávy, potom se vztahuje pouze na tuto část.

4.1.4 Hlavička Content-ID

Volitelná hlavička, která nese jednoznačnou identifikaci obsahu zprávy.

4.1.5 Hlavička Content-Description

Obsahuje popisné informace o přenášené zprávě, například název obrázku, který je jako tělo posílán.

4.1.6 Hlavička Content-Disposition

Určuje, jsou-li přenášená data určena k automatickému zobrazení, nebo zda mají být připojena jako příloha. První případ se vyjadřuje hodnotou *inline*, druhý hodnotou *attachment*. Dále může hlavička obsahovat ještě další parametry:

- `filename` – jméno souboru
- `creation-date` – datum vytvoření
- `modification-date` – datum poslední modifikace
- `read-date` – datum posledního čtení
- `size` – velikost přenášeného souboru

4.2 Standardní kódovací algoritmy

4.2.1 Quoted-Printable

Toto kódování je určeno pro data, která z větší části obsahují ASCII znaky. Ty nechává nezměněné, takže výsledek je i bez dekódování z velké části čitelný.

[1] hodnoty 8bit, 7bit a binary nepředstavují žádné kódování, jsou užitečné jako indikace typu dat

Pravidla kódování

- byty s desítkovou hodnotou 33 – 60 a 62 – 126 včetně jsou nahrazeny odpovídajícími znaky ASCII. Tedy jak již bylo řečeno, znaky ASCII se zakódováním nezmění (mimo znak `=`).
- ostatní byty se nahrazují znakem `=` následovaným šestnáctkovou hodnotou bytu. A to i samotný znak `=`.
- byty s hodnotou 9 a 30 jsou nahrazeny znaky tabulátor a mezera. Nesmí být na konci řádku.
- konec řádku je vyjádřen jako `CRLF`.
- zakódovaný řádek musí mít maximálně 76 znaků. Pokud je řádek delší, použije se měkký konec řádku, tj. znak `=` a konec řádku. Tedy například první věta této odrážky se zakóduje jako
`zak=F3dovan=FD =F8=E1dek mus=ED m=EDt maxim=E1ln=EC 76 znak=F9.`

4.2.2 Base64

Kódování Quoted-Printable v nejhorším případě prodlouží kódovaný text na trojnásobek. V běžném textu tato situace nenastává, ale pro obecná data se kvůli tomu nehodí. K tomu je určeno kódování Base64, které prodlouží původní data pouze právě o třetinu.

Kódovací algoritmus používá tabulku Base64 o 64 znacích (velká a malá písmena anglické abecedy, číslice, + a /) a znak `=`. Funguje tak, že se na data dívá jako na proud šestic bitů a postupně každé takové šestici přiřadí odpovídající znak z tabulky Base64 ($2^6 = 64$). Problém nastává v okamžiku, kdy počet bytů není dělitelný třemi. Pak není možné rozdělit data na šestice bytů. Tato potíž je řešena tak, že se zbylé 2 (resp. 4) bity doplní nulami a na konec zakódovaného řetězce se jako signál přidá znak `=` (resp. 2 znaky `=`).

5 Protokol SMTP a jeho rozšíření

5.1 SMTP

Pro předání zprávy se používá protokol *SMTP – Simple Mail Transfer Protocol*, který je založený na architektuře klient-server. SMTP je poměrně jednoduchý protokol. Jednotlivé příkazy jsou čtyřznaková slova v kódu ASCII (podobně jako u protokolu FTP). Klient navazuje spojení protokolem TCP se serverem na portu 25. Klient vkládá do takto vytvořeného kanálu příkazy a server odpovídá odpovědí obsahující trojciferný stavový kód a následovaný textovým popisem chyby. Příkaz může být následován parametrem, který je oddělen mezerou. Příkaz je zakončen koncem řádku.

5.1.1 Přehled příkazů SMTP

`HELO` *klient*	Klient se představuje serveru jménem počítače (používá se na počátku dialogu).
`MAIL FROM:` *odesilatel*	Specifikuje odesílatele.
`RCPT TO:` *příjemce*	Specifikuje příjemce (opakuje se pro každého příjemce).

`DATA`	Klient se chystá poslat zprávu (žádá o schválení).
`RSET`	Abnormální ukončení spojení (a zahození již zaslaných údajů).
`SEND FROM:` *odesilatel*	Obdoba příkazu `MAIL`, ale zpráva má být zobrazena na terminálu příjemce. Nepoužívá se.
`SOML FROM:`*odesilatel*	Obdoba příkazu `MAIL`, ale zpráva má být zobrazena na terminálu nebo uložena do poštovní schránky příjemce. Nepoužívá se.
`SAML FROM:` *odesilatel*	Obdoba příkazu `MAIL`, ale zpráva má být zobrazena na terminálu a také uložena do poštovní schránky příjemce. Nepoužívá se.
`VRFY` *adresa*	Dotaz, zda-li příjemce zná uvedenou adresu. Server vrací plné jméno uživatele a jeho přesnou poštovní identifikaci.
`EXPN` *adresa*	Obdoba `VRFY`, ale umí pracovat nejen s jednotlivými uživateli, ale i s celými seznamy uživatelů.
`HELP [` *příkaz*`]`	Nápověda
`QUIT`	Ukončení spojení.
`TURN`	Přepnutí role serveru a klienta. Tento příkaz se nepoužívá kvůli nebezpečí, že kdokoliv bez jakékoliv autentizace může přijmout ze serveru frontu mailů.

5.1.2 Přehled stavových kódů SMTP

500	Syntaktická chyba, příkaz unrecognized (Toto zahrnuje chyby jako command line too long)
501	Syntaktická chyba v parametrech nebo argumentech
502	Příkaz neimplementován
503	Špatné pořadí příkazů
504	Parametr (u příkazu) neimplementován
211	System status, or system help reply
214	Help message (Information on how to use the receiver or the meaning of a particular non-standard command; this reply is useful only to the human user)
220 <*domain*>	Služba připravena
221 <*domain*>	Service closing transmission channel
421 <*domain*>	Service not available, closing transmission channel (This may be a reply to any command if the service knows it must shut down)
250	Requested mail action okay, completed
251	User not local; will forward to ¡forward-path¿
252	Cannot VRFY user, but will accept message and attempt delivery
450	Requested mail action not taken: mailbox unavailable (e.g., mailbox busy)
550	Requested action not taken: mailbox unavailable (e.g., mailbox not found, no access, or command rejected for policy reasons)
451	Requested action aborted: error in processing

551	User not local; please try ¡forward-path¿
452	Requested action not taken: insufficient system storage
552	Requested mail action aborted: exceeded storage allocation
553	Requested action not taken: mailbox name not allowed (e.g., mailbox syntax incorrect)
354	Start mail input; end with ¡CRLF¿.¡CRLF¿
554	Transaction failed (Or, in the case of a connection-opening response, "No SMTP service here")

5.2 Rozšíření SMTP

Do protokolu SMTP byla přidána řada rozšíření, která jsou souhrnně označována jako *ESMTP – Extensions SMTP*. Zpětná kompatibilita je zajištěna příkazem `EHLO`, který nahrazuje příkaz `HELO`. Klient totiž příkazem `EHLO` začíná dialog a pokud server ESMTP nepodporuje, vrátí chybové hlášení, že se klient spletl v názvu příkazu. Pro klienta je to zpráva, že má užívat pouze SMTP bez rozšíření a pokračuje příkazem `HELO`. Pokud server ESMTP podporuje, pak na příkaz `EHLO` odpoví seznamem všech podporovaných rozšíření. (povinné příkazy jsou `HELO`, `MAIL`, `RCPT`, `DATA`, `RSET`, `VRFY`, `NOOP` a `QUIT`, mezi volitelné příkazy patří `SEND`, `SOML`, `SAML`, `EXPN`, `HELP`, `TURN` a mezi rozšířené `ETRN`, `BDAT`).

5.2.1 Popis některých rozšíření

PIPELINING	Klient může odesílat další SMTP příkazy bez čekání na odpověď na předchozí příkaz.
8BITMIME	Umožňuje přenos zpráv MIME obsahujících osmibitová data.
SIZE	Specifikace délky zprávy. Server v odpovědi na příkaz `EHLO` vrací toto rozšíření s parametrem určujícím maximální akceptovatelnou délku zprávy. Klient jej může použít jako parametr u příkazu `MAIL` a sdělit tak, jak dlouhou zprávu posílá.
CHUNKING	Umožňuje přenos binárních zpráv. Místo příkazu `DATA` se používá příkaz `BDAT`.
ETRN	Tento příkaz umožňuje provoz poštovním serverům, které nemají stálé připojení (např. malé firmy). Jejich zprávy jsou zůstavají na serveru poskytovatele a ten je doručí, až když se připojí. Příkaz `ETRN` zajišťuje, že server zašle zprávy ihned po připojení a ne až na ně přijde řada při procházení frontou. Používá se s jediným parametrem, tím je doména, pro kterou má server začít prohledávání fronty.
DSN	Toto rozšíření umožňuje notifikaci doručení zprávy. Rozšiřuje příkaz `MAIL` o parametry `RET` a `ENVID`. `RET` určuje, zda v notifikační zprávě vrátit celý obsah původní zprávy (`RET=FULL`) nebo jen záhlaví (`RET=HDRS`) a `ENVID` identifikuje zprávu. `DSN` dále rozšiřuje `RCPT` o parametry `NOTIFY` a `ORCPT`. `NOTIFY` specifikuje podmínku, za které se má notifikace generovat (`NEVER` – nikdy, `SUCCES` – při doručení, `FAILURE` – při selhání doručení, `DELAY` – při zpoždění doručení). `ORCPT` určuje původního adresáta.

6 POP a IMAP

POP a IMAP jsou protokoly sloužící ke stažení zpráv ze schránky na poštovním serveru na osobní počítač.

6.1 POP

POP – Post Office Protocol je jednoduchý protokol určený pro práci offline. Je to vlastně jednosměrný komunikační kanál, kterým jsou k uživateli, po předchozí autentizaci, dopraveny zprávy nacházející se v jeho mailboxu.

Základní autentizace je jménem a heslem uživatele. V případě, že autentizace proběhne bez problému, přechází komunikace do transakčního stavu, kdy klient pracuje se zprávami na serveru. Během transakčního stavu jsou veškeré změny ještě odvolatelné, na závěr relace přejde klient do stavu UPDATE, kdy se změny na serveru provedou trvale.

Server na příkaz vrací odpověď začínající buď znakem +, pokud je vše v pořádku, nebo znakem –, což signalizuje chybu.

6.1.1 Seznam příkazů

1. Autentizace

   - `USER` *jméno* – Zadání jména uživatele.
   - `PASS` *heslo* – Zadání hesla (příkaz je volitelný).
   - `QUIT` – Ukončení spojení.

2. Transakční stav

   - `STAT` – Vrací počet zpráv a celkovou velikost schránky.
   - `LIST` – Vrací seznam zpráv v poštovní schránce, u každé zprávy je uvedeno pořadové číslo a velikost.
   - `RETR` *číslo_zprávy* – Stažení zprávy.
   - `DELE` *číslo_zprávy* – Smazání zprávy ve schránce na serveru.
   - `NOOP` – Prázdný příkaz.
   - `RSET` – Obnoví zprávy zrušené během relace
   - `TOP` *číslo_zprávy počet_řádků* – Vypíše počátek zprávy.
   - `QUIT` – Přechod do stavu UPDATE a následné ukončení relace.

6.2 IMAP

Protokol *IMAP – Internet Message Acces Protocol* je dokonalejší obdobou POP. Základní rozdíl je ten, že veškeré zprávy zůstávají uloženy na serveru. Uživatel může zprávy třídit do složek, prohledávat a mazat bez toho, aby musel celý obsah mailboxu přenášet na svůj počítač. Může si třeba přečíst jen tělo mailu bez nutnosti stahovat přes síť i všechny objemné MIME přílohy. Novější revize IMAPu umějí synchronizovat uživatelův lokální mailbox s poštou na vzdáleném serveru. Největší předností IMAPu oproti POP je, že uživatel může ke své poště přistupovat z více různých míst a všude bude mít k dispozici stejné zprávy.

Příkazy IMAP jsou stejně jako u POP v ASCII, ale jejich odlišnost od příkazů POP je značná. Je to způsobeno tím, že v IMAP může být zadávána řada příkazů a odpovědi na ně mohou přicházet v libovolném pořadí. Proto klient příkazy čísluje a server do odpovědi přidá číslo příkazu, na který odpovídá. Identifikace řetězce je obecně řetězec (nemusí to tedy být ani číslo) a je zcela v rukou klienta.

Server odpovídá dvěma druhy odpovědí:

1. Nečíslovanými odpověďmi, které mají na místě čísla příkazu-odpovědi hvězdičku `*`. Tyto nečíslované řádky nesou informaci, na kterou se klient dotazoval
2. Číslovanými odpověďmi, které se skládají z čísla příkazu a výsledku. Výsledek může nabývat těchto hodnot:
   (a) `OK` – příkaz proběhl úspěšně
   (b) `NO` – příkaz proběhl neúspěšně
   (c) `BAD` – chybný příkaz
   (d) `PREAUTH` – server ihned po navázání spojení sděluje, že je klient přihlášen jako konkrétní uživatel bez nutnosti autentizace (neobvyklé)
   (e) `BYE` – server ukončuje spojení

Práce se serverem IMAP probíhá ve třech módech:

1. Neautentizovaný stav
2. Autentizovaný stav
3. Otevřená schránka

V neautentizovaném stavu ještě není klient oprávněn nic dělat, pouze se přihlásit. V autentizovaném stavu klient pracuje s jednotlivými poštovními schránkami (složkami). A konečně, když některou schránku otevře, může zpracovávat už i jednotlivé zprávy, které obsahuje. Mezi autentizovaným stavem a otevřenou poštovní schránkou se může klient samozřejmě volně přepínat pomocí příkazů `SELECT` a `EXAMINE` (otevření schránky), respektive `CLOSE` (uzavření schránky, přechod do autentizovaného stavu).

6.2.1 Příkazy neautentizovaného stavu

V neautentizovaném stavu je uživatel bezprostředně po připojení. Pokud nechce ihned server opustit (příkazem `LOGOUT`, který je společný pro všechny tři módy), může se přihlásit pomocí jednoho z příkazů `LOGIN` a `AUTHENTICATE`. Příkazem `LOGIN` se přihlašuje běžným způsobem pomocí jména a hesla, zatímco pro jiné přihlašovací mechanismy slouží příkaz `AUTHENTICATE`. Klient jako parametr tohoto příkazu předá autentizační mechanizmus, který chce použít, a podle něj pak pokračuje další komunikace.

6.2.2 Příkazy autentizovaneho stavu

V autentizovaném stavu už je možností více. Klient může vytvářet přejmenovávat a mazat poštovní schránky, nechat si je vypsat či o nich získávat podrobnější informace. Příkazy v autentizovaném stavu jsou následující:

- `CREATE` *název_schránky* – vytvoření schránky.
- `DELETE` *název_schránky* – zrušení schránky.
- `RENAME` *původní_název nový_název* – přejmenování schránky.
- `LIST` *cesta název_schránky* – výpis schránek. V názvu schránky mohou být znaky hromadného výběru `*` a `%`. Hvězdička rozvíjí vše, a procento pouze název schránky, ne adresářovou strukturu.
- `SUBSCRIBE` *název_schránky* – označí soubor jako poštovní schránku.
- `SUBSCRIBE` *název_schránky* – zruší označení poštovní schránky.
- `LSUB` – obdoba příkazu `LIST`; vypisuje pouze označené schránky.
- `STATUS` *název_schránky (seznam_položek)* – vypíše informace (podle zadaného seznamu položek) o poštovní schránce, aniž by byla otevřena. Položky jsou:
  - `MESSAGES` – počet zpráv ve schránce.
  - `RECENT` – počet zpráv s příznakem \Recent.
  - `UIDNEXT` – číslo zprávy, které bude přiřazeno nově příchozí zprávě ve schránce.
  - `UIDVALIDITY` – jednoznačná identifikace schránky.
  - `UNSEEN` počet zpráv bez příznaku \Seen.
- `SELECT` *název_schránky* – přechod do režimu "Otevřená poštovní schránka" (server v odpovědi vypíše informace o schránce)
- `EXAMINE` *název_schránky* – totéž jako `SELECT`, ale schránka je otevřena pouze pro čtení

6.2.3 Příkazy v módu "Otevřená poštovní schránka"

Než popíšeme příkazy, je třeba se zmínit o identifikaci zpráv, která je pro některé z nich nezbytná. Každá poštovní schránka má jednoznačnou identifikaci a pro určení zpráv se používá sekvenční číslování od jedničky v rámci schránky. Při vymazání zprávy dochází současně i k přečíslování.

- `COPY` *rozsah_zpráv schránky* – zkopíruje zprávy z otevřené poštovní schránky do schránky uvedené jako druhý parametr
- `SEARCH` *kritéria* – vyhledá v poštovní schránce zprávy. Vyhledává je podle kritérií uvedených jako parametr. Výchozí spojka mezi nimi je AND, explicitně lze samozřejmě použít samozřejmě i spojku OR a negaci. Kritérií je celá řada, zahrnují například výskyt řetězce v těle, specifikaci odesilatele, výskyt řetězce v dané hlavičce, rozmezí velikosti zprávy, rozmezí data odeslání apod.
- `FETCH` *rozsah_zpráv* `[`*část*`]` – stáhne zprávu nebo její část. Argument část je nepovinný, ale také atraktivní. Specifikuje jaká část zprávy má být stažena. Jeho hodnota může být např.:

- BODY[*sekce*][2] – server vrátí obsah sekce uvedené v hranatých závorkách. Sekce jsou HEADER (všechny hlavičky), HEADER.FIELDS (hlavičky jejichž jména dodána v kulatých závorkách), HEADER.FIELDS.NOT (hlavičky, kromě těch, jejichž jména jsou dodána v kulatých závorkách), MIME (MIME hlavičky) a TEXT (text zprávy).
- BODY[*sekce*]<od.do> – totéž jako předchozí s tím, že do špičatých závorek je navíc uvedeno rozmezí přenesených dat (v bytech).
- BODY.PEEK[*sekce*]<od.do> – opět totéž, jennevrací zprávy s příznakem \Seen
- FLAGS – vrací říznaky uvedených zpráv
- BODYSTRUCTURE – vrací strukturu zprávy (jednotlivé MIME části)

* STORE *rozsah_zpráv režim (seznam_příznaků)* – slouží ke změně příznaků zpráv. Nastavením příznaku \Delete se zpráva připraví k odstranění (odstraněna je až při použití příkazu EXPUNGE). Argument *režim* je jedno z klíčových slov FLAGS, +FLAGS a -FLAGS. +FLAGS příznaky přidává, -FLAGS odebírá a FLAGS nastavuje právě uvedené.
* EXPUNGE – smazání položek označených příznakem \Delete.
* CLOSE – přechod do režimu "Autentizovaný stav".

[2]Pozor, hranaté závorky zde neoznačují nepovinný parametr, ale jsou součástí zápisu

Reference

[1] Dostálek, L.: *Velký průvodce protokoly TCP/IP: Bezpečnost*
Computer Press, Praha 2001

[2] Petříček, M.: *Lehký úvod do elektronické pošty*
`http://www.petricek.cz/clanky/lehky_uvod_do_el_posty.html`, 2001

[3] Freed, N., Borenstein, N., Moore, K., Klensin, J., Postel, J.:
Multipurpose Internet Mail Extensions (RFC 2045 – 2049)
`ftp://ftp.isi.edu/in-notes/rfc2045.txt - rfc2049.txt`, 1996

[4] Klensin, J.: *Simple Mail transfer Protocol (RFC 2821)*
`ftp://ftp.isi.edu/in-notes/rfc2821.txt`, 2001

[5] Crispin, M.: *Internet Message Access Protocol – Version 4rev1 (RFC 3501)*
`ftp://ftp.isi.edu/in-notes/rfc3501.txt`, 2003