



Klient IMAP s podporou TLS

Autor: Milan Jakubec (xjakub41)

Brno, 17. listopadu 2024

Obsah

1	Teoretický základ	3
1.1	Protokol IMAP	3
1.1.1	Způsob komunikace	3
1.1.2	LOGIN/LOGOUT	3
1.1.3	SELECT	3
1.1.4	FETCH	4
1.1.5	SEARCH	4
1.2	TLS	4
1.3	Internet Message Format	5
2	Návrh aplikace	6
2.1	Požadavky	6
2.2	Návrh	6
3	Popis implementace	6
3.1	Zvolená řešení	7
3.1.1	Timeouty	7
3.1.2	Možnosti transformace emailů	7
3.1.3	Pojmenování emailových souborů	7
3.1.4	Prevence znovustahování emailů	8
3.1.5	Reakce na odpovědi serveru	8
3.1.6	Volba příznaku pro přepínač -n	8
4	Základní informace o programu	9
5	Návod na použití	9
5.1	Formát autentizačního souboru	10
5.2	Příklad použití	10
6	Testování	10
6.1	Práce s parametry	10
6.1.1	Test: Chybějící povinný parametr	10
6.1.2	Test: Nadbytečný parametr	11
6.1.3	Test: Použití -c bez použití -T	11
6.2	Autentizace	11
6.2.1	Test: Chybějící soubor autentizace	11
6.2.2	Test: Chybné autentizační údaje	12
6.3	Reakce na neočekávané chování	12
6.3.1	Test: Připojení k neexistujícímu serveru	12
6.3.2	Test: Náhlé ukončení spojení	12

6.4	Komunikace přes TLS	13
6.4.1	Test: Stažení všech zpráv ze serveru s TLS	13
6.5	Šťastné scénáře	13
6.5.1	Test: Stažení všech zpráv ze serveru	13
6.5.2	Test: Stažení pouze hlavičkových zpráv	14
6.5.3	Test: Stažení pouze nových zpráv	14
6.5.4	Test: Žádné zprávy k synchronizaci	14
Bibliografie		14

1 Teoretický základ

1.1 Protokol IMAP

RFC 3501[1] definuje Internet Message Access Protocol (dále IMAP) jako síťový protokol, který umožňuje klientovi přístup k emailovým zprávám a manipulaci s nimi na serveru. V tomto dokumentu je rozebírána verze protokolu z tohoto RFC, tedy IMAP4rev1, nicméně je vhodné zmínit, že od roku 2021 je tato verze nahrazena verzí IMAP4rev2, specifikovanou v RFC 9051.

IMAP je obousměrný protokol, který kopíruje poštu ze serveru na stranu klienta, přičemž původní kopie zpráv zůstávají na serveru zachovány. Změny provedené na těchto zprávách, jako třeba přečtení či přesunutí do jiné složky, jsou nicméně na serveru reflektovány.[4] Stažené zprávy na lokální zařízení jsou uživateli k dispozici, i když není k dispozici připojení k internetu.

1.1.1 Způsob komunikace

IMAP je založen na textových příkazech, které klient odesílá serveru. Každý příkaz začíná unikátním identifikátorem (tzv. tag), který umožňuje klientovi sledovat odpovědi serveru, za nímž následuje samotný příkaz. Dále protokol definuje tři klíčová slova, která bývají zahrnuta v odpovědích: OK, NO a BAD, přičemž první zmiňované typicky značí úspěšně provedený příkaz, druhé zmiňované značí neúspěch, a třetí typicky nějakou chybu či špatně zadané argumenty. Následující příkazy jsou využívány v rámci tohoto projektu a z hlediska protokolu IMAP patří mezi nejdůležitější.

1.1.2 LOGIN/LOGOUT

Příkaz LOGIN slouží k autentizaci uživatele pomocí uživatelského jména a hesla. Po ukončení práce s protokolem se používá příkaz LOGOUT k uzavření spojení.

```
A001 LOGIN username password
A002 LOGOUT
```

1.1.3 SELECT

Slouží k volbě emailové schránky, v rámci níž bude přístupováno k emailovým zprávám. Pokud je příkaz úspěšně vykonán, předtím, než server klientovi vrátí OK, musí předem zaslat příznaky např. definovaných příznaků schránky, informaci o počtu zpráv ve schránce aj. Příklad níže demonstrovuje zvolení schránky INBOX.

A003 SELECT INBOX

1.1.4 FETCH

Získávání specifických částí zpráv, například hlaviček, těl nebo příloh. Pomocí tohoto příkazu je možné získat buďto jednotlivou zprávu, či celý seznam. Příklad níže demonstruje získání všech celých zpráv, aniž by byl nastaven příznak SEEN (přečteno).

A004 FETCH 1:* (BODY.PEEK[])

1.1.5 SEARCH

Vyhledávání zpráv podle určitých kritérií, například klíčových slov nebo data. Může sloužit například i k získání všech unikátních identifikátorů emailových zpráv. Příklad níže demonstruje vyhledání zpráv počínaje specifikovaným datem.

A005 SEARCH SINCE 1-Jan-2024

1.2 TLS

Protokol Transport Layer Security (dále TLS) je v RFC 8446[2] definován jako standardní kryptografický protokol zajišťující bezpečnou komunikaci v síti. Byl navržen jako nástupce protokolu SSL (Secure Sockets Layer) a primárně poskytuje:

- **Důvěrnost:** Šifrování přenášených dat, které brání odposlechu třetími stranami.
- **Autentizaci:** Ověření identity serveru a/nebo klienta pomocí digitálních certifikátů.
- **Integritu:** Detekci jakékoliv manipulace s daty během přenosu.

TLS se běžně používá v aplikacích, jako jsou webové prohlížeče, e-mailové služby, VoIP a jiné. Aktuální verzí je **TLS 1.3**.

Postup navázání TLS spojení

Navázání TLS spojení (tzv. *TLS handshake*) probíhá ve třech hlavních fázích:

1. **Dohoda o šifrovacích algoritmech a parametrech:** Klient a server si vymění seznam podporovaných šifrovacích algoritmů a dohodnou se na společném.
2. **Výměna klíčů:** Používá se asymetrická kryptografie (např. algoritmus Diffie-Hellman) k bezpečné výměně klíčů.
3. **Autentizace a zahájení šifrované komunikace:** Server se identifikuje pomocí certifikátu a po úspěšné autentizaci začíná šifrovaný přenos dat.

Využitím této ochranné vrstvy je zajištěno bezpečné navázání spojení, které chrání citlivá data před odposlechem a případnými útoky třetích stran.

1.3 Internet Message Format

Emaily přenášené pomocí IMAP protokolu jsou standardizovány formátem Internet Message Format (dále jen IMF), jež je definován v RFC 5322[3] a definuje pravidla pro organizaci hlaviček zpráv, těla zprávy a jejich syntaxi. IMF slouží jako základní protokol pro výměnu e-mailů v internetových sítích a je využíván nejen protokolem IMAP, nýbrž i dalšími, jako jsou například SMTP a POP3. Právě díky této standardizaci je možné, aby spolu mohly různé emailové služby komunikovat.

Formát IMF rozděluje emailovou zprávu na dvě hlavní části: hlavičku zprávy, která obsahuje strukturované informace o odesílateli, příjemci a další metadata (skládající se z řádků formátu "klíč: hodnota"), a tělo zprávy, které obsahuje samotný obsah emailu.

2 Návrh aplikace

2.1 Požadavky

Úkolem tohoto projektu je napsat program `imapcl`, který umožní čtení elektronické pošty pomocí protokolu IMAP4rev1. Program po spuštění stáhne zprávy uložené na serveru a uloží je do zadaného adresáře (každou zprávu zvlášť). Na standardní výstup vypíše počet stažených zpráv. Pomocí datečných parametrů je možné funkcionalitu měnit.

2.2 Návrh

Na základě specifikovaných požadavků byl zpracován následující návrh:

Bude vytvořen program v jazyce C++ z důvodu možnosti využití tříd a větší bezpečnosti při práci s pamětí. Sestávat bude ze tří hlavních modulů, z čehož jeden bude pro zpracování argumentů a jejich uložení do definované datové struktury, jeden pro zpracování emailové zprávy a uložení do souboru, a třetí pro vytvoření abstrakce nad síťovými operacemi typu připojení a zaslání/čtení zpráv serveru. Pomocné funkce pro vykonávání specifických operací, jako například zpracovávání přihlašovacích údajů ze souboru, budou v samostatné třídě určené pro tento účel. Samotný program se potom bude vykonávat v souboru `Program`.

3 Popis implementace

Program klienta začne tím, že zpracuje argumenty použité při spuštění do struktury nazvané *ParsedArgs*. Následně se pomocí instance třídy *IMAPClient* pokusí připojit k serveru s případným využitím TLS. Při úspěšném navázání spojení jako první odešle klient příkaz `login`, načež si přečte odpověď na tuto zprávu. Implementace funguje tak, že okamžitě po odeslání zprávy je načtena odpověď ze serveru do proměnné datového typu `string`. Pakliže proběhne autentizace v pořádku, odešle klient následně zprávu s volbou `schránky` a zkontroluje aktuálnost `UID` pro daný server. V případě, že `UID` se změnilo, smaže klient své lokální emailové zprávy a soubor s `UID` uloží. Toto se používá pro udržení aktuálnosti informací. Následně je dle specifikovaných argumentů sestavený `FETCH` příkaz, pomocí kterého jsou získány všechny emaily. Klient žádá o všechny emaily najednou, díky čemuž předchází tomu, že by například došlo k výpadku komunikace během iterativního získávání zpráv. Po získání odpovědi na `FETCH` příkaz je tato odpověď zpracována a rozčleněna do C++ vektoru `stringů`, který je následně v

cyklu procházen, je vytvořena instance třídy *EmailMessage*, a pomocí metod této třídy je emailová zpráva uložena do souboru. Následně je dle požadavku ze zadání na standartní výstup vypsáno, jaký počet emailových zpráv byl stažen.

3.1 Zvolená řešení

Tato sekce v krátkosti vysvětluje některá zvolená řešení na konkrétní problémy, jejichž řešení nebylo specifikováno v zadání projektu.

3.1.1 Timeouty

Aby se předešlo tomu, že by klient donekonečna čekal, je implementováno timeoutování klienta. Jednak na samotném připojení, přičemž doba čekání byla nastavena na 5 sekund. Tato doba je v běžném provozu často používané nastavení, které poskytuje dostatečný časový prostor pro překlenutí případných síťových zpoždění či zpracování dat. Zahrnuje výraznou bezpečnostní rezervu. Tento timeout je nastaven i při očekávání odpovědi ze serveru. Pro implementaci této mechaniky bylo využito funkce *select()* a struktury *timeval*.

3.1.2 Možnosti transformace emailů

Zadání neupřesňovalo, zda umožňovat nějaké transformace (například již stažený email degradovat pouze na hlavičkový soubor v případě znovupoužití programu s parametrem *-h*). Tato implementace klienta umožňuje napřed stáhnout hlavičkové soubory emailů, a při dalším použití klienta a vynechání argumentu *-h* se hlavičkové soubory smažou a stáhne se kompletní emailová zpráva. Opačným směrem tato transformace autorovi nedává logický smysl, a z toho důvodu není v této implementaci umožněna.

3.1.3 Pojmenování emailových souborů

Volba jména pro emailový soubor představovala poměrně velký problém. V prvotních verzích implementace bylo zamýšleno využití serveru a message ID, nicméně message ID je parametr, který může v jistých případech být prázdný. Nakonec pro jméno souboru je zvolena kombinace kanonického názvu serveru, názvu schránky a UID dané zprávy. Poslední jmenované klient využívá pro zjištění, jaká UID emailů již má lokálně k dispozici. Mimo jiné tato varianta byla zvolena proto, aby v jedné lokální složce bylo možno ukládat emaily z více různých serverů, pro což by například kombinace emailu odesílatele a UID zprávy nemusela být vhodná.

3.1.4 Prevence znovustahování emailů

Aby práce s emailovými zprávami byla efektivní, je implementovaný mechanismus, který předchází zbytečnému znovustahování emailů, které již jsou k dispozici a tedy opravdu docházelo pouze k "synchronizaci" klienta a serveru. Pro tyto účely je zůžitkováno UID emailové zprávy, jak je již vysvětleno v předchozí podsekci. Klient si získá dle uložených souborů seznam lokálně dostupných UID, vyžádá si seznam všech UID od serveru, tyto seznamy porovná a následně si vyžádá ty, které mu lokálně chybí. Rozlišuje i UID souborů, které jsou pouze hlavičkové, a které jsou celé emaily, aby v případě, že je to žádáno, byly tyto hlavičkové soubory smazány a byl místo nich stažen kompletní email.

3.1.5 Reakce na odpovědi serveru

Klient je připraven adekvátně reagovat na různé odpovědi či zprávy serveru. Může se stát, že server na některou z klientových zpráv zareaguje například klíčovým slovem BAD, případně NO. V případě, že se toto stane, klient tuto odpověď zaregistruje a slušným způsobem komunikaci ukončí společně s vypsáním chybové hlášky. Další situace, která může nastat, je, že server během odpovídání na zprávu klienta odešla tzv. "untagged" odpověď, tedy odpověď, které chybí odpovídající značka (např. A001). Klient si dává pozor i na takovéto zprávy, které pro účely tohoto projektu jednoduše ignoruje a přeskakuje, nicméně nezpůsobí u klienta neočekávané chování.

3.1.6 Volba příznaku pro přepínač -n

Zadání projektu definuje použití přepínače "-n" tak, že bude pracovat (číst) pouze s novými zprávami. Jako nejideálnější volba se v kombinaci s příkazem SEARCH jeví příznak NEW, který efektivně vyhledává zprávy, které mají nastavený příznak Recent a zároveň nemají nastavený příznak Seen. Funkčně je vlastně toto ekvivalentní s "(RECENT UNSEEN)".

4 Základní informace o programu

Program slouží jako konzolový klient pro práci s protokolem IMAP (přesněji IMAPrev1). Po spuštění stáhne zprávy uložené na serveru a uloží je do zadaného adresáře (každou zprávu zvlášť). Na standardní výstup vypíše počet stažených zpráv. Pomocí dodatečných parametrů je možné funkcionalitu měnit, například stáhnout pouze hlavičkové soubory, či pouze nové emailové zprávy. Taktéž je možné využít variantu šifrované komunikace. Program je určen pro uživatele, kteří by chtěli jednoduše synchronizovat emaily mezi serverem a lokálním počítačem bez závislosti na komerčních emailových klientech.

Pro vytvoření programu byl zvolen programovací jazyk C++ s využitím standardní knihovny a knihoven pro práci se síťovou komunikací (např. iostream, iomanip, openssl, sys, netinet, arpa aj.)

5 Návod na použití

Program se spouští z příkazového řádku s následující syntaxí:

```
imapcl server [-p port] [-T [-c certfile] [-C certaddr]] [-n] [-h]
-a auth_file [-b MAILBOX] -o out_dir
```

Pořadí parametrů je libovolné. Níže je uveden popis jednotlivých parametrů:

- **Povinné parametry:**

- **server** – Název serveru (IP adresa nebo doménové jméno), ke kterému se má program připojit.
- **-a auth_file** – Cesta k souboru obsahujícímu autentizační údaje pro příkaz LOGIN. Podrobnosti o formátu souboru viz níže.
- **-o out_dir** – Výstupní adresář, do kterého se uloží stažené emailové zprávy.

- **Volitelné parametry:**

- **-p port** – Specifikuje port, na kterém je server dostupný. Výchozí hodnota závisí na použití parametru **-T**.
- **-T** – Zapíná šifrování (TLS). Pokud není tento parametr uveden, program TLS používat nebude.
- **-c certfile** – Cesta k souboru s certifikáty.

- **-C certaddr** – Adresář, kde se vyhledávají certifikáty pro ověření platnosti SSL/TLS certifikátu serveru. Výchozí hodnota je `/etc/ssl/certs`.
- **-n** – Program bude pracovat pouze s novými e-maily, tedy takovými, které nebyly ještě přečteny a jsou nedávné.
- **-h** – Umožňuje stahování pouze hlaviček e-mailových zpráv bez jejich obsahu.
- **-b MAILBOX** – Specifikuje schránku na serveru, se kterou má program pracovat. Výchozí hodnota je `INBOX`.

5.1 Formát autentizačního souboru

Konfigurační soubor s autentizačními údaji, specifikovaný parametrem `-a`, musí obsahovat uživatelské jméno a heslo v následujícím formátu:

```
username = jmeno  
password = heslo
```

Soubor by měl být ve formátu unixového textového souboru s ukončením znakem nového řádku.

5.2 Příklad použití

Níže je uveden příklad spuštění programu:

```
imapcl imap.exampleserver.com -T -a authfile -o ./emails
```

Tento příkaz připojí program k serveru na portu 993 (je využito TLS), načte autentizační údaje ze souboru `authfile`, pracuje se složkou `INBOX` a stáhne e-maily do adresáře `./emails`.

6 Testování

6.1 Práce s parametry

6.1.1 Test: Chybějící povinný parametr

Vstup:

```
./imapcl imap.pobox.sk -T -o roundcube
```

Výstup:

```
Error: Parametrers -a (auth_file) a -o (out_dir) are required!  
Usage: ./imapcl server [-p port] [-T [-c certfile] [-C certaddr]]  
[-n] [-h] -a auth_file [-b MAILBOX] -o out_dir
```

6.1.2 Test: Nadbytečný parametr

Vstup:

```
./imapcl imap.pobox.sk imap.pobox.sk -T -a pobox_auth -o maildir
```

Výstup:

```
Error: Expected exactly one server argument. Found 2 arguments.  
Usage: ./imapcl server [-p port] [-T [-c certfile] [-C certaddr]]  
[-n] [-h] -a auth_file [-b MAILBOX] -o out_dir
```

6.1.3 Test: Použití -c bez použití -T

Vstup:

```
./imapcl imap.pobox.sk -c -a pobox_auth -o roundcube
```

Výstup:

```
Error: Parametr -c (certfile) is only used with -T (TLS).  
Usage: ./imapcl server [-p port] [-T [-c certfile] [-C certaddr]]  
[-n] [-h] -a auth_file [-b MAILBOX] -o out_dir
```

6.2 Autentizace

6.2.1 Test: Chybějící soubor autentizace

Vstup:

```
./imapcl imap.pobox.sk -T -a missingfile -o outputdir
```

Výstup:

```
Failed to open authfile: missingfile
```

6.2.2 Test: Chybné autentizační údaje

Vstup:

```
./imapcl imap.pobox.sk -T -a pobox_auth -o outputdir
```

Výstup:

```
Error: Authentication failed.
```

6.3 Reakce na neočekávané chování

6.3.1 Test: Připojení k neexistujícímu serveru

Vstup:

```
./imapcl madeupimap.pobox.sk -T -a pobox_auth -o outputdir
```

Výstup:

```
Error: Failed to resolve hostname.
```

6.3.2 Test: Náhlé ukončení spojení

V tomto manuálním testu využiji lokálního mock python imap serveru, na který klienta připojím, a následně ze strany serveru ukončím spojení.

Vstup:

```
./imapcl 127.0.0.2 -a auth -o outputdir
```

Výstup:

```
Error: Connection closed by server.
```

6.4 Komunikace přes TLS

6.4.1 Test: Stažení všech zpráv ze serveru s TLS

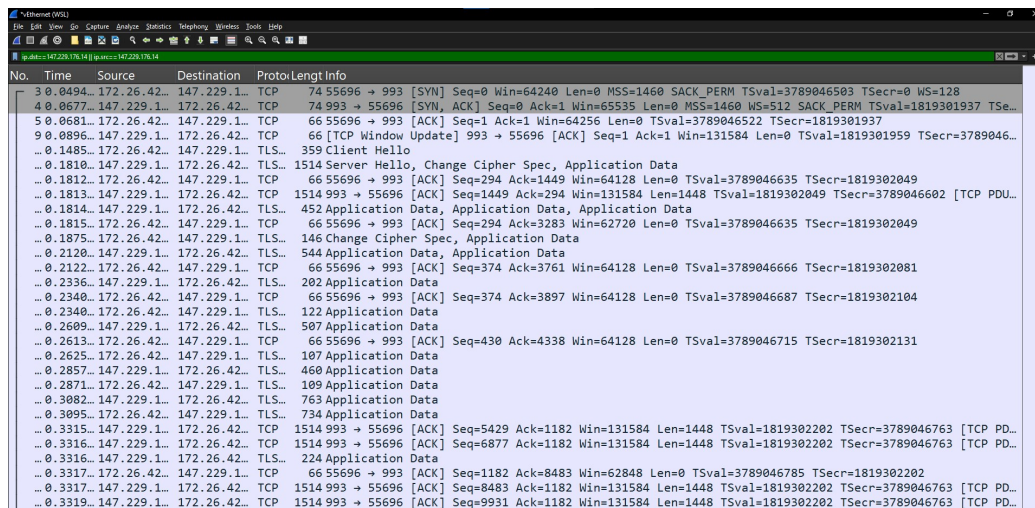
Vstup:

```
./imapcl eva.fit.vutbr.cz -T -a roundcube_auth -o roundcube
```

Výstup:

Downloaded 125 messages from mailbox INBOX

Zachycená Wireshark komunikace:



No.	Time	Source	Destination	Protocol	Length	Info
3	0.0494...	172.26.42...	147.229.1...	TCP	74	55696 → 993 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3789046503 TSecr=0 WS=128
4	0.0677...	147.229.1...	172.26.42...	TCP	74	993 → 55696 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=512 SACK_PERM TSval=1819301937 TSecr=...
5	0.0681...	172.26.42...	147.229.1...	TCP	66	55696 → 993 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3789046522 TSecr=1819301937
9	0.0896...	147.229.1...	172.26.42...	TCP	66	[TCP Window Update] 993 → 55696 [ACK] Seq=1 Ack=1 Win=131584 Len=0 TSval=1819301959 TSecr=3789046...
...
...	TLS	359	Client Hello
...	TLS	1514	Server Hello, Change Cipher Spec, Application Data
...	TCP	66	55696 → 993 [ACK] Seq=294 Ack=1449 Win=64128 Len=0 TSval=3789046635 TSecr=1819302049
...	TCP	1514	993 → 55696 [ACK] Seq=1449 Ack=294 Win=131584 Len=1448 TSval=1819302049 TSecr=3789046602 [TCP PD...
...	TLS	452	Application Data, Application Data, Application Data
...	TCP	66	55696 → 993 [ACK] Seq=294 Ack=3283 Win=62720 Len=0 TSval=3789046635 TSecr=1819302049
...	TLS	146	Change Cipher Spec, Application Data
...	TLS	544	Application Data, Application Data
...	TCP	66	55696 → 993 [ACK] Seq=374 Ack=3761 Win=64128 Len=0 TSval=3789046666 TSecr=1819302081
...	TLS	202	Application Data
...	TCP	66	55696 → 993 [ACK] Seq=374 Ack=3897 Win=64128 Len=0 TSval=3789046687 TSecr=1819302104
...	TLS	122	Application Data
...	TLS	507	Application Data
...	TCP	66	55696 → 993 [ACK] Seq=430 Ack=4338 Win=64128 Len=0 TSval=3789046715 TSecr=1819302131
...	TLS	107	Application Data
...	TLS	460	Application Data
...	TLS	109	Application Data
...	TLS	763	Application Data
...	TLS	734	Application Data
...	TCP	1514	993 → 55696 [ACK] Seq=5429 Ack=1182 Win=131584 Len=1448 TSval=1819302202 TSecr=3789046763 [TCP PD...
...	TCP	1514	993 → 55696 [ACK] Seq=6877 Ack=1182 Win=131584 Len=1448 TSval=1819302202 TSecr=3789046763 [TCP PD...
...	TLS	224	Application Data
...	TCP	66	55696 → 993 [ACK] Seq=1182 Ack=8483 Win=62848 Len=0 TSval=3789046785 TSecr=1819302202
...	TCP	1514	993 → 55696 [ACK] Seq=8483 Ack=1182 Win=131584 Len=1448 TSval=1819302202 TSecr=3789046763 [TCP PD...
...	TCP	1514	993 → 55696 [ACK] Seq=9931 Ack=1182 Win=131584 Len=1448 TSval=1819302202 TSecr=3789046763 [TCP PD...

Obrázek 1: Snímek síťové komunikace při využití TLS

6.5 Šťastné scénáře

6.5.1 Test: Stažení všech zpráv ze serveru

Vstup:

```
./imapcl eva.fit.vutbr.cz -T -a roundcube_auth -o roundcube
```

Výstup:

Downloaded 125 messages from mailbox INBOX

6.5.2 Test: Stažení pouze hlavičkových zpráv

Vstup:

```
./imapcl eva.fit.vutbr.cz -T -a roundcube_auth -o roundcube -h
```

Výstup:

```
Downloaded 125 messages (headers only) from mailbox INBOX
```

6.5.3 Test: Stažení pouze nových zpráv

Vstup:

```
./imapcl eva.fit.vutbr.cz -T -a roundcube_auth -o roundcube -n
```

Výstup:

```
No new messages found.
```

6.5.4 Test: Žádné zprávy k synchronizaci

Vstup:

```
./imapcl eva.fit.vutbr.cz -T -a roundcube_auth -o roundcube
```

Výstup:

```
No new messages to synchronize.
```

Bibliografie

Odkazy

- [1] Mark Crispin. *Internet Message Access Protocol - Version 4rev1*. Accessed: 2024-11-16. Břez. 2003. URL: <https://www.rfc-editor.org/rfc/rfc3501>.

- [2] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. Accessed: 2024-11-16. Srp. 2018. URL: <https://www.rfc-editor.org/rfc/rfc8446>.
- [3] Pete Resnick. *Internet Message Format*. Accessed: 2024-11-16. Srp. 2008. URL: <https://datatracker.ietf.org/doc/html/rfc5322>.
- [4] soucet. *Rozdíl mezi protokoly IMAP a POP3*. Accessed: 2024-11-16. Srp. 2021. URL: <https://support.mozilla.org/cs/kb/rozdil-mezi-protokoly-imap-pop3>.