

# Linux prakticky ako server / 21.časť

V minulej časti sme si rozdelili poštu do niekoľkých základných variantov a vysvetlili princíp činnosti, výhody a nevýhody a spôsob určenia. Teraz sa začneme jednotlivým variantom venovať podrobnejšie.

## Variant koncového klienta

Pôvodne som nechcel nastavenie tohto klienta popisovať, lebo sa mi zdalo, že to už všetci dobre vedia. Ale po pár konkrétnych otázkach mojim potenciálnym žiakom som zistil, že nie všetci nováčikovia sú v tomto nastavení úplne doma.

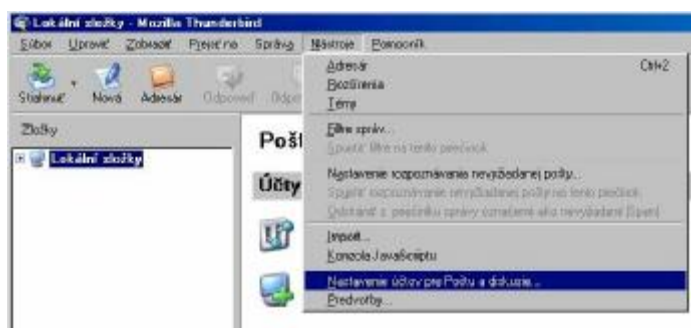
A aj preto a hlavne pre poriadok si to teraz ukážeme.

Nech pre ilustráciu máme dané tieto údaje:

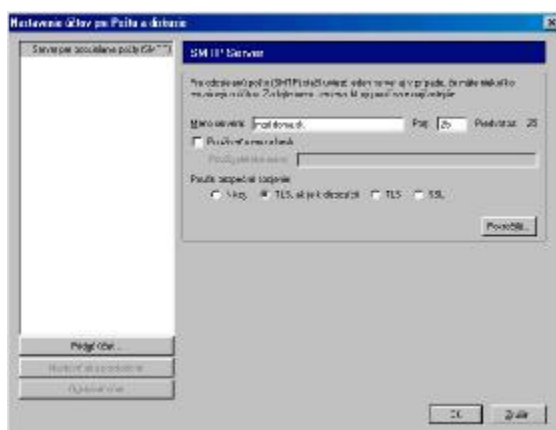
Účet u poskytovateľa poštových služieb s poštovým serverom *mail.doma.sk*, meno účtu *leonka* (zároveň je to prihlasovacie meno) a prihlasovacie heslo *7CeV3ro*. A schránka sa bude vyberať protokolom POP3.

Tieto spomenuté údaje teraz nastavíme na strane klienta v programe MUA. Pre náš príklad použijeme čoraz obľúbenejší program **Mozilla Thunderbird**, ktorý je dostupný pre obidva operačné systémy – MS Windows aj Linux (a dokonca je dostupný v rôznych jazykových mutáciách).

Spustíme teda „hromového vtáka“ (*Thunder* = *hrom*, *bird* = *vták*) a v menu *Nástroje* vyberieme položku *Nastavenie účtov pre Poštu a diskusie...* (obr.č.1):

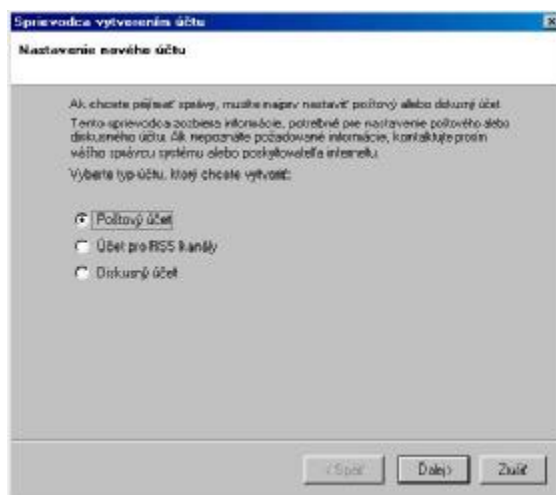


Zobrazí sa okno s názvom *Nastavenie účtov pre Poštu a diskusie*. V ľavej časti okna klikneme na položku s menom *Server pre odosielanie pošty (SMTP)*. V pravej časti sa zobrazí okno, kde nastavíme požadované hodnoty – obr.č.2:

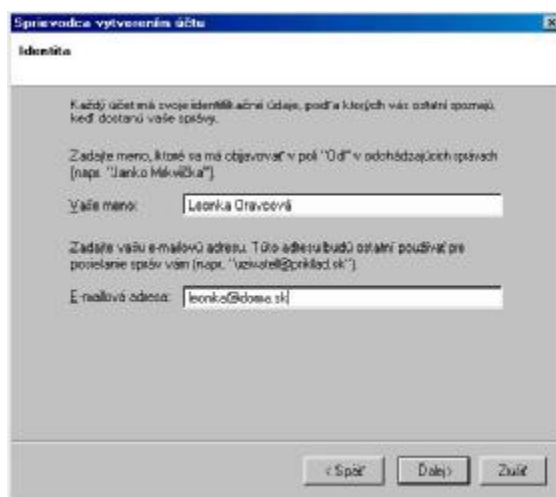


Napíšeme meno SMTP servera, ktoré nám stanovil náš poskytovateľ poštovej služby. Všimnime si, že port je defaultne nastavený na hodnotu 25 (a nie je zatiaľ dôvod ho meniť). V prípade, že SMTP server vyžaduje overenie z dôvodu ochrany pred šírením spamu, zaklikneme položku *Používať meno a heslo* a meno vypíšeme. Potom klikneme na tlačidlo *Pridať účet...*

Otvorí sa nové okno *Spríevodcu vytvorením účtu* – obr.č.3:

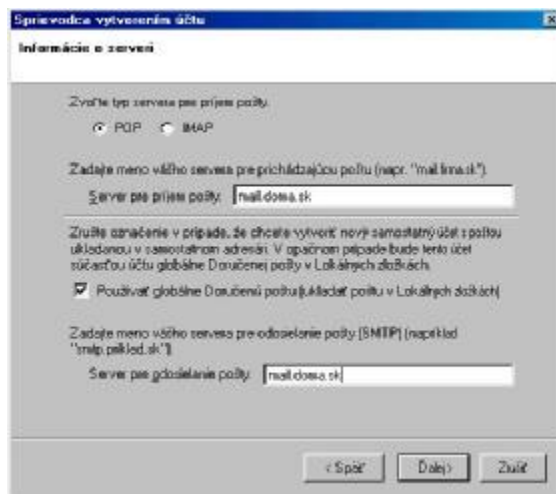


Vyberieme položku *Poštový účet* a klikneme na tlačidlo *Ďalej>*. Prejdeme do ďalšieho okna *Spríevodcu* – obr.č.4:



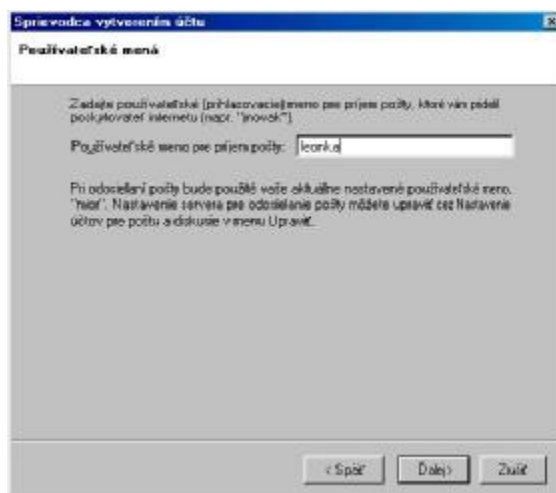
Zadáme meno, pod ktorým sa budeme oznamovať okolitému svetu, napríklad Leonka Oravcová a zadáme aj pridelenú emailovú adresu, teda v našom prípade leonka@doma.sk. Klikneme na tlačidlo *Ďalej>*.

V nasledujúcom okne zvolíme typ servera pre prichádzajúcu poštu a zadáme adresu poštového servera – obr.č.5:

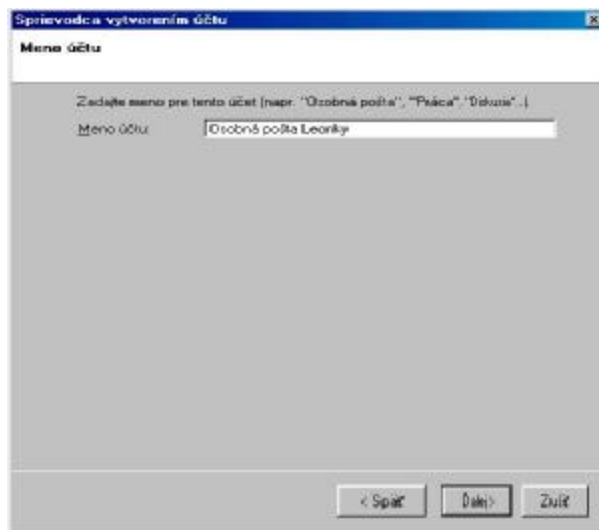


Povedali sme si, že budeme používať protokol POP3 a tak zaklikneme tento protokol. Po zadaní mena servera klikneme na *Ďalej>*.

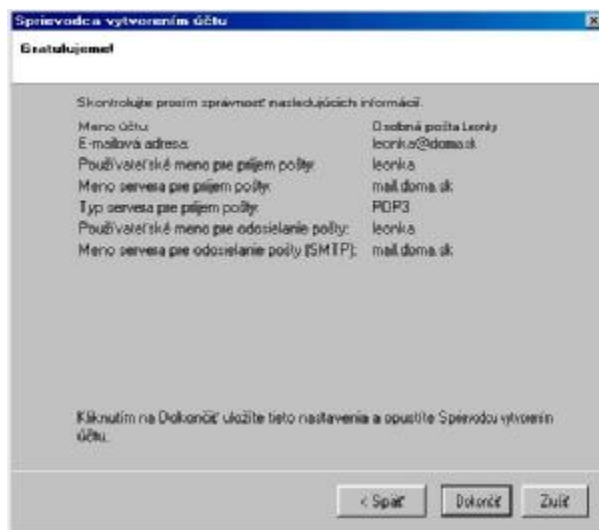
Nakoniec zadáme prihlasovacie meno – to býva totožné s názvom účtu. Nezabudnime, že si ho nemôžeme svojvoľne zmeniť, pretože sa viaže na nastavenie poštovej schránky na strane poštového servera! (obr.č.6):



Po kliknutí na *Ďalej>* nám zostáva už len pomenovať účet, ako sa nám bude zobrazovať v programe MUA – obr.č.7:



Po odkliknutí sa objaví posledná záverečná stránka *Sprievodcu*, kde sa objavia sumárne údaje nášho nastavenia – obr.č.8:



Klikneme na tlačidlo *Dokončiť* a môžeme prijímať poštu!  
Hotovo!

*Poznámka:*

*Sprievodca nastavením účtu sa môže spustiť aj sám, a to hneď po nainštalovaní programu.*

### Protokoly

Doteraz sme sa na poštu pozerali ako obyčajní používatelia. Nevadí, lebo aj to treba!

Teraz sa však na poštu pozrieme z iného pohľadu.

Opustíme poštový program MUA a k pošte pristúpime na nižšej úrovni. Priamo z klávesnice počítača budeme zadávať príkazy poštovému serveru tak, ako keby to robil program MUA. Na to budeme používať dva už nám známe protokoly – SMTP a POP3.

Predstavme si, že sme Leonka a chceme poslať správu sami sebe (má to tú výhodu, že si odskúšame jej odoslanie a zároveň si vyskúšame jej príjem).

**Pozor!**

*Tu uvedené príklady sú imaginárne. Mená, názvy serverov, heslá a IP adresy sú vymyslené. Preto pri skúšaní dosadzte vám známe mená účtov, poštových serverov, IP adresy a heslá!*

## SMTP protokol

Na prístup k serveru SMTP použijeme program *telnet*. Je to dávno, čo sa tento program v prostredí počítačových sietí intenzívne používal, v dnešnej dobe oknoidných prostredí o ňom bežný používateľ počítača vôbec nevie a používajú ho už iba administrátori. My však chceme byť dobrými administrátormi a tak sa s týmto programom zoznámime.

### Telnet

*V prehistorických dobách, keď sálový počítač zaberá jednu veľkú sálu, sa s ním pracovalo pomocou terminálov. Boli to v podstate „vynesené“ (rozumej vzdialené) obrazovky a klávesnice. Časom sa terminály nahradili inteligentnejšími zariadeniami, v dnešnej dobe osobnými počítačmi. Avšak systém terminálovej práce s niektorými aplikáciami zostal dodnes. Preto bolo potrebné emulovať terminálový režim. Preto vznikol program telnet.*

*Telnet je teda program, ktorý emuluje terminálový režim v textovom režime.*

Ešte predtým, než sa spojíme so SMTP serverom a napíšeme správu, môžeme zistiť (ak to ešte nevieme), kto je príjemcom pošty pre doménu *doma.sk*.

V riadku shellu zadáme príkaz:

```
[root@rubin root]# host -t mx doma.sk
```

Na obrazovke sa objaví výpis:

```
doma.sk mail is handled by 10 mail.doma.sk
```

Takto sme zistili (a ak sme to už vedeli, tak sme si overili), že pre doménu *doma.sk* je príjemcom pošty server *mail.doma.sk*.

A teraz sa spojíme so serverom *mail.doma.sk* pomocou *telnetu*. Vieme, že protokol SMTP načúva na porte 25. Takže zadáme príkaz

```
[root@rubin root]# telnet mail.doma.sk 25
```

Na obrazovke sa objaví výpis:

```
Trying 217.17.236.7...
Connected to mail.doma.sk (217.17.236.7).
Escape character is '^]'.
220 mail.doma.sk ESMTP Postfix
```

Tým zostane relácia v *telnete* otvorená a my môžeme zadávať príkazy (v *telnete*, takže neexistuje žiadny prompt, iba kurzor):

Najprv sa so serverom pozdravíme (nemusíme, ale niektoré servery si pozdrav vyslovene vyžadujú!):

```
helo doma.sk
```

A server nám odpovie:

```
250 mail.doma.sk
```

A teraz začneme písať správu. Najprv napíšeme, od koho správa je. Na to slúži príkaz **mail from:** (aj s dvojbodkou!). Takže pokračujeme:

```
mail from: leonka@doma.sk
```

(Meno odosielateľa by sme si mohli aj vymyslieť. Aj v skutočnej pošte sa stáva, že sa niekto skrýva za iným alebo neskutočným menom).

Ak sme zadali príkaz správne, server nám odpovie kódom 250:

```
250 ok
```

Teraz určíme adresáta – príjemcu pošty. Na to slúži príkaz **rcpt to:**.  
Zadáme:

```
rcpt to: leonka@doma.sk
```

Server správnosť potvrdí kódom 250:

```
250 ok
```

Teraz nastáva čas na obsah správy. Začína sa príkazom **data**.  
Zadáme:

```
data
```

Server odpovie, že môžeme zadávať obsah a že po skončení máme stlačiť *Enter*, na nový riadok napísať znak . (bodku) a zase stlačiť *Enter*:

```
354 End data with <CR><LF>.<CR><LF>
```

Do obsahu správy patrí aj názov – subject – správy.  
Zadáme:

```
Subject: Test poslania spravy
```

Po stlačení Enteru server neodpovie, ale očakáva zadanie samotného obsahu správy.  
Preto pokračujeme:

```
Toto je skuska poslania spravy. Mior.
```

Keď dopíšeme obsah správy, stlačíme Enter, napíšeme bodku a zase stlačíme Enter. Tým sme dali serveru na vedomie, že sme ukončili editáciu správy.  
Server to potvrdí oznamom:

```
250 ok: queued as D11E45AF15
```

Správu sme napísali, teraz slušne ukončíme spojenie.  
Zadáme:

```
quit
```

A server odpovie:

```
221 Bye  
Connection closed by foreign host.
```

Po ukončení spojenia so serverom sa vrátime do príkazového riadku shellu.

Čože je to za zvláštne číslo D11E45AF15?

Pod týmto číslom je správa uložená do frontu a odoslaná do sveta.

Aby sme to mali ucelenejšie, na výpise č.10 je obsah obrazovky pri komunikácii so SMTP serverom pomocou telnetu. Text klienta je modrou farbou, odpovede SMTP servera červenou farbou (v skutočnosti je na obrazovke všetko čierno biele – podľa typu použitého telnetu):

```
[root@rubin root]# telnet mail.doma.sk 25  
Trying 217.17.236.7...  
Connected to mail.doma.sk (217.17.236.7).
```

```
Escape character is '^]'.
220 mail.doma.sk ESMTP Postfix
helo doma.sk
250 mail.doma.sk
mail from: leonka@doma.sk
250 ok
rcpt to: leonka@doma.sk
250 ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: Test poslania spravy
Toto je skuska poslania spravy. Mior.
.
250 ok: queued as D11E45AF15
quit
221 Bye
Connection closed by foreign host.
```

A verme, že naša správa bude zaslaná.

### POP3 protokol

Odteraz by sa mala správa nachádzať v poštovej schránke s názvom *leonka* na serveri *mail.doma.sk*. Aby sme si danú poštu prečítali, využijeme protokol POP3.

A zase využijeme na pripojenie k serveru program telnet. POP3 protokol načúva na porte 110, preto zadáme príkaz:

```
[root@rubin root]# telnet mail.doma.sk 110
```

Server nadviaže spojenie a odpovie napríklad takto:

```
Trying 217.17.236.7...
Connected to mail.doma.sk (217.17.236.7).
Escape character is '^]'.
+OK <5364.1112459116@mail.doma.sk>
```

Server je v očakávaní príkazov v telnetovej relácii.

Ako prvé sa musíme serveru predstaviť. Na to slúži príkaz **user meno\_používateľa**. Zadáme príkaz:

```
user leonka
```

a server odpovie:

```
+OK
```

Teraz sa musíme serveru preukázať, že sme to naozaj my. Na to sa používa autorizácia heslom. Preto zadáme:

```
pass 7CeV3ro
```

(Pozor! Heslo sa vypisuje v otvorenej podobe, žiadne hviezdičky, ako sme zvyknutí. Pracujeme v primitívnom programe telnet, nie v inteligentnom MUA!)

Ak je heslo správne, odpoveď servera je:

```
+OK
```

Ak sme nezadali správne meno alebo heslo, server odpovie:

```
-ERR Username/password does not match
Connection closed by foreign host.
```

Predstavme si, že sme meno a heslo zadali správne a preto pokračujeme.  
Chceme si pozrieť, koľko máme správ v našej schránke. Na to slúži príkaz **list**.  
Zadáme:

**list**

A server odpovie:

```
1 1925
2 2058
.
```

Z výpisu vidíme, že máme dve správy. Prvá, označená číslom 1 má veľkosť 1925 bajtov, druhá - označená číslom 2 má veľkosť 2058 bajtov.

Tá bodka na poslednom riadku symbolizuje ukončenie výpisu.

Ak si chceme pozrieť obsah správy, použijeme príkaz **retr**.

Pre výpis správy pod číslom 1 zadáme:

**retr 1**

Na obrazovke sa objaví telo správy:

```
Return-Path: <leonka@doma.sk>
X-Envelope-To: leonka@doma.sk
Delivered-To: doma-sk__-leonka@doma.sk
Received: (qmail 39072 invoked from network); 2 Apr 2005 18:34:31 +0200
Received: from unknown (HELO localhost) (217.17.236.7)
  by mail.doma.sk with SMTP
  for <leonka@doma.sk>; 2 Apr 2005 18:34:31 +0200
Received: from mail.doma.sk [213.151.236.161])
  by server (Postfix) with ESMTP id D11E45AF15
  for <leonka@doma.sk>; Sat, 2 Apr 2005 18:34:31 +0200 (CEST)
Subject: Test poslanca spravy
Message-ID: <001201c537a2$86079280$020aa8c0@doma.sk>
Date: Sat, 2 Apr 2005 18:39:22 +0200
From: leonka@doma.sk
To: leonka@doma.sk

Toto je skuska poslanca spravy.Mior.

.
```

Všimnime si! Okrem subjektu a samotného obsahu správy sa vypísali ďalšie znaky. Tým hovoríme obálka alebo hlavička správy. Ak by správa obsahovala aj iné formátovacie znaky, napríklad príkazy HTML jazyka alebo dokonca binárnu prílohu, v tomto výpise by sme ich uvideli „v plnej nahote“.

Keď však takúto správu prijme program MUA, zobrazí sa používateľovi v rozumnom formáte!

Ak chceme určitú správu vymazať z poštovej schránky, použijeme príkaz **dele**.

Zadáme:

**dele 1**

Server odpovie známym:

```
+OK
```

Prácu v telnete ukončíme zadaním

**quit**



a server odpovie:

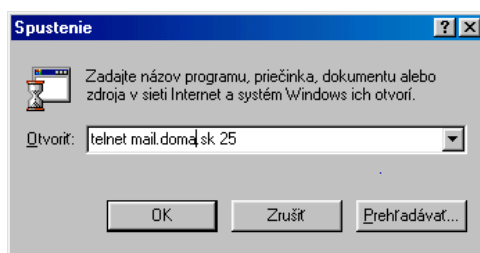
```
+OK  
Connection closed by foreign host.
```

Tak ako pri SMTP protokole, tak aj tu si to dajme dokopy (server červenou, klient modrou farbou):

```
[root@rubin root]# telnet mail.doma.sk 110  
Trying 217.17.236.7...  
Connected to mail.doma.sk (217.17.236.7).  
Escape character is '^J'.  
+OK <5364.1112459116@mail.doma.sk>  
user leonka  
+OK  
pass 7CeV3ro  
+OK  
list  
1 1925  
2 2058  
.  
retr 1  
  
--- správa---  
  
dele 1  
+OK  
quit  
+OK  
Connection closed by foreign host.
```

Po ukončení spojenia sa vrátíme do shellu.

Program telnet môžeme použiť aj v prostredí MS Windows. Stačí, ak v menu **Štart – Spustiť** zadáme príkaz *telnet mail.doma.sk 25* – obr.č.9:



A spustí sa telnet v prostredí MS Windows – obr.č.10:



Možno si povieme, že načo tak komplikovane tvoriť a čítať poštu, keď existujú hotové klientské programy MUA. Lenže keď budeme zisťovať, prečo nám pošta nechodí, ako vylúčime, kde je chyba? Preto použijeme telnetovú reláciu. Ak tu bude všetko v poriadku, už budeme vedieť, že problém je niekde inde, napríklad v prenose alebo v MUA. A práve toto budeme veľmi často používať pri tvorbe pošty v našom systéme. Nabudúce sa budeme intenzívne venovať programu **postfix**.

# Linux prakticky ako server / 22.časť

V minulej časti sme si ukázali, ako sa konfiguruje koncový klient poštovej služby MUA. Zároveň sme si skúsili používať príkazy protokolov SMTP a POP3 v príkazovom riadku konzoly.

Dnes pristúpime ku konfigurácii poštového servera variantu s nepriamym doručovaním pošty.

## Nepriame doručovanie pošty

Už sme si hovorili, že variant nepriameho doručovania elektronickej pošty je vhodný pre malé a stredné firmy alebo školy. Ak si spomenieme, hovorili sme, že aj tento variant môže mať niekoľko modifikácií, záleží, či chceme okrem redistribúcie pošty vykonávať aj iné operácie s poštou, ako je preposielanie, antivírová a antispamová ochrana, archivácia či iné skutočne špeciálne neštandardné činnosti.

Treba spomenúť, že tento spôsob je najvhodnejší v kombinácii s doménovým košom.

Keďže nie všetci majú možnosť služby doménového koša, ukážeme si, ako využívať tento variant na klasickom účte, ktorý máme vytvorený u svojho poskytovateľa elektronickej poštovej služby.

Sám používam tento systém aj doma, pretože mám niekoľko poštových účtov na Internete a nechce sa mi ich z historických (či skôr nostalgických) dôvodov meniť. Takto si sťahujem poštu zo všetkých sólo účtov na svoj poštový server, na ňom vykonávam triedenie a potom si ich vyberám poštovým klientom MUA.

Modifikácia variantu so sólo účtom je veľmi podobná modifikácii s doménovým košom, je tu len niekoľko zmien.

A aby sme si to všetci precvičili, obidve modifikácie si ukážeme.

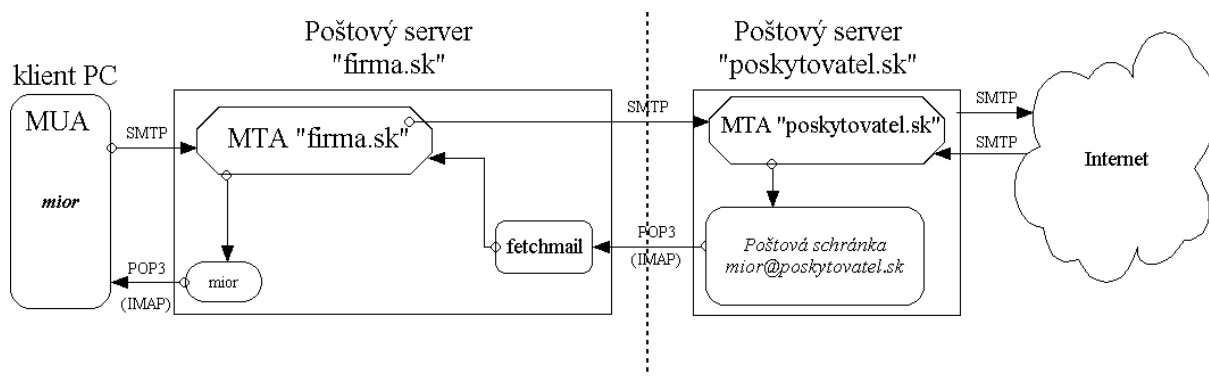
## Nepriame doručovanie pošty z jednoduchého poštového účtu.

Nech máme u imaginárneho poskytovateľa internetových služieb s menom „poskytovatel.sk“ vytvorenú poštovú schránku s názvom *mior@poskytovatel.sk*, s menom *mior* a heslom *sk78cv3*.

Na priamy výber elektronickej pošty by nám samozrejme stačil aj obyčajný poštový klient MUA s nastavením, ktoré sme si ukázali v minulej časti.

Ale keďže my chceme v budúcnosti s touto poštou ešte pred jej čítaním v MUA vykonávať určité operácie, vytvoríme si vlastný poštový úrad s názvom FIRMA.

Schématický náčrt je na obrázku č.1:



Náš poštový server s názvom *firma.sk* bude akoby vklinený medzi poskytovateľa a klientsky program MUA. Veci naľavo od prerušovanej čiary budú u nás doma, veci napravo budú na Internete.

## Princíp činnosti

Úlohou nášho poštového servera FIRMA je prijímať poštu od MUA klientskeho počítača a túto správu preposlať pomocou protokolu SMTP a cestou nadriadeného poštového servera POSKYTOVATEL do sveta príjemcovi. V prípade, že je pošta určená príjemcovi na našom lokálnom serveri, bude táto správa uložená do jeho schránky. Pošta, adresovaná z Internetu nám bude klasicky uložená na serveri poskytovateľa v poštovej schránke *mior@poskytovatel.sk*. Z tadiaľ ju budeme vyberať pomocou programu fetchmail protokolom POP3, ktorý ju po vyzdvihnutí odovzdá programu postfix. Ten sa zachová štandardne a poštu uloží do príslušnej schránky *mior*.

Táto schéma ako aj popisovaná konfigurácia nie je závislá od spôsobu pripojenia do Internetu alebo k poskytovateľovi poštovej služby. Môže sa jednať o pripojenie pevnou linkou, pomocou ADSL, Wi-Fi alebo pomocou vytáčaného modemového spojenia. Všetky možnosti sme si už rozoberali v predchádzajúcich častiach seriálu a preto sa k nim teraz nebudeme vracieť.

(Tí, ktorí nemajú doma možnosť poštovej služby, nech skúsia pripojenie v škole. Dnes už každá škola má pripojenie do Internetu a tak požiadajte správcu siete o poskytnutie takejto služby).

Na sprevádzkovanie poštového servera potrebujeme okrem fungujúceho počítača s nainštalovaným a sprevádzkovaným Linuxom, pripojeným do Internetu tieto programy:

- Ø MTA program **postfix**
- Ø Program **fetchmail**
- Ø Program s protokolom POP3

Všetky tri programy sú štandardnou súčasťou každej distribúcie Linuxu a tak ich isto nájdeme na príslušných inštalačných cd-čiečkách. Ak by sme predsa len zatúžili po najnovších verziách, tak program postfix sa nachádza na internetovej stránke [www.postfix.org](http://www.postfix.org) a program fetchmail na stránke <http://www.catb.org/~esr/fetchmail/>.

### Postfix

Postfix je v poslednej dobe najobľúbenejší poštový server, ktorý plne nahrádza robustný, ťažko konfigurovateľný a veľmi zraniteľný prehistorický program *sendmail*.

Postfix je pomerne jednoduchý na konfiguráciu a pritom veľmi výkonný SMTP server.

Skladá sa z niekoľkých modulov, kde každý modul vykonáva presne špecifikovanú činnosť. Takto je zabezpečená prehľadnosť činnosti a zdrojových kódov, ale aj bezpečnosť. Chyba v jednom module nezastaví činnosť celého poštového servera, ale len daného modulu.

Konštitúcia postfixu veľmi pripomína program *inetd* a konfiguračný súbor jednotlivých procesov s názvom *master.cf* pripomína *inetd.conf* (preberali sme si to! Spomeňte si!).

Jednotlivé demony – služby sa spúšťajú ako samostatné procesy. Ich počet je daný konfiguráciou.

Sú to tieto služby:

- Ø **smtpd** – počúva na stanovenom porte a prijíma SMTP požiadavky. Prijaté poštové správy sú presmerované na démona *cleanup*
- Ø **pickup** – čaká na lokálne vytvorené správy a presmeruje ich obsah na démona *cleanup*
- Ø **cleanup** – spracováva prijatú správu, pridáva chýbajúcu hlavičku, vkladá ju do frontu prijatých správ a informuje démona *qmgr* o jej príchode
- Ø **qmgr** – čaká na prijaté správy a zabezpečuje ich doručenie. Spôsob doručenia určuje démon *trivial-rewrite*
- Ø **trivial-rewrite** – prepisuje adresu do normovanej formy. Démon pripája meno domény k lokálnym správam bez jej uvedenia a podobne. Určuje, čo sa stane so správou, ako a kam sa má doručiť na základe adresy
- Ø **local** – doručuje správy do lokálnych schránok na serveri
- Ø **smtp** – SMTP klient programu postfix. Doručuje správy z poštového frontu, ktoré sú určené pre iné poštové servery.

### Dôležité adresáre a súbory programu postfix

V štandardnej inštalácii servera postfix sa používajú tieto adresáre a súbory:

- Ø */etc/postfix/* – adresár s konfiguračnými súbormi
- Ø */etc/postfix/master.cf* – hlavný konfiguračný súbor demonov, nastavenia jednotlivých služieb
- Ø */etc/postfix/main.cf* – samotný konfiguračný súbor s nastaveniami doručovania a odosielania pošty
- Ø */etc/aliases* – súbor s fiktívnymi adresami – aliasmi používateľov
- Ø */var/spool/postfix/* – adresár s dátovými štruktúrami postfixu počas jeho prevádzky. Tu sa nachádzajú práve doručované, prijímané alebo nedoručiteľné správy.
- Ø */etc/postfix/\** - ostatné súbory – tabuľky

### Obsluha programu postfixu

Program sa ovláda pomocou klasických príkazov na ovládanie služieb Linuxu.

Na spustenie postfixu môžeme použiť príkaz

```
[root@rubin root]# /etc/init.d/postfix start
```

Na zastavenie môžeme použiť príkaz

```
[root@rubin root]# /etc/init.d/postfix stop
```

a na prípadný reštart programu použijeme príkaz

```
[root@rubin root]# /etc/init.d/postfix restart
```

Ak urobíme do konfigurácie počas behu programu nejaké zásahy a zmeny, nemusíme program reštartovať, ale stačí, ak vykonáme znovunačítanie konfigurácie príkazom

```
[root@rubin root]# /etc/init.d/postfix reload
```

Ak si nie sme istí, či sú konfiguračné súbory upravené správne bez syntaktických (nie logických!) chýb, môžeme ich otestovať príkazom

```
[root@rubin root]# /etc/init.d/postfix check
```

Ak nastala na našej sieti chyba pri doručovaní správ a na serveri uviazli neodoslané správy, po odstránení chyby (napr. v spojení do sveta) môžeme všetky čakajúce správy odoslať príkazom

```
[root@rubin root]# /etc/init.d/postfix flush
```

V distribúciách založených na RedHate môžeme použiť aj utilitu **service** v tvare

```
[root@rubin root]# service postfix start|stop|restart|reload|abort|flush|check|status
```

Ak sa chceme pozrieť na obsah frontu správ, čakajúcich na odoslanie, použijeme príkaz

```
[root@rubin root]# mailq
```

## Konfigurácia

Základná konfigurácia sa vykonáva v súbore **main.cf**.

Tento súbor obsahuje množstvo direktív, ktorými sa ovláda celý poštový server. Mnohé z týchto direktív sú už prednastavené autorom programu a sú naozaj veľmi dobre zdokumentované v samotnom súbore.

V adresári s dokumentáciou postfixu, napr. */usr/share/doc/postfix/* (niekde *len /usr/doc/postfix/*) sa nachádza podadresár */samples/*, v ktorom sú uložené rôzne príklady konfigurácie SMTP servera pre jednotlivé prípady použitia. Aby sme porozumeli uvedeným príkladom, musíme si povedať niečo o samotných direktívach.

Každá direktíva má tvar *parameter = hodnota*. Ak je hodnotou zoznam viacerých položiek, tieto sa oddeľujú medzerou alebo čiarkou. Niektoré direktívy nemajú skutočnú hodnotu, ale sa odkazujú na direktívu definovanú predtým. Riadky začínajúce medzerou sú pokračovaním predchádzajúceho dlhého riadku.

## Najdôležitejšie direktívy

Aj keď je väčšina direktív prednastavená a má defaultnú hodnotu, predsa len existujú direktívy, ktoré je VHODNÉ nami nadefinovať, aby server pracoval správne.

Sú to tieto direktívy:

### myhostname

Určuje plné meno servera, na ktorom server SMTP beží. Spravidla to je meno linuxového servera.

Príklad:

```
myhostname = mail.firma.sk
```

Toto meno v prípade variantu s nepriamym doručovaním nemusí byť verejne známe v Internete a preto si ani nemusíme nechávať registrovať doménu *firma.sk*. Bude to len naša lokálna doména, zakrytá serverom *poskytovatel.sk*.

### mydomain

Určuje doménové meno servera. Štandardne je to hodnota premennej *myhostname* bez prvej položky.

Príklad:

```
mydomain = firma.sk
```

**myorigin**

Určuje meno servera v odchádzajúcej pošte z tohto servera. Toto meno sa pripája za meno lokálneho používateľa, ak nebola špecifikovaná doména. Štandardne sa používa odkaz na hodnotu direktívy *myhostname*.  
Príklad:

*myorigin = \$myhostname*

**inet\_interfaces**

Určuje zoznam adries, na ktorých sa má prijímať pošta. Štandardne je to na všetkých.  
Príklad:

*inet\_interfaces = all*

**mydestination**

Tu sa definujú doménové adresy, ktoré postfix považuje za lokálne a pre ktoré prijíma poшту. Ak sa tu pomýlime, náš server nebude dobre prijímať poшту.

Príklad:

*mydestination = \$myhostname, localhost.\$mydomain, \$mydomain, localhost*

**mynetworks**

Obsahuje zoznam IP adries siete, ktorú postfix považuje za dôveryhodnú. Iba z takto označenej siete sa môže odosielať pošta. Býva to rozsah našej siete.

Príklad:

*mynetworks = 192.168.10.0/24, 127.0.0.0/8*

Ak je tento parameter nesprávny, postfix odmieta odosielať poшту.

**relay\_domains**

Určuje doménové adresy, pre ktoré postfix umožňuje *relay* – preposielanie pošty. Štandardná hodnota je rovnaká ako v *mydestination*

Príklad:

*relay\_domains = \$mydestination*

**relay\_host**

Server SMTP postfix sa snaží doručiť poшту tak, že si prečíta adresu príjemcu a v záznamoch DNS si zistí, kto je zodpovedný za príjem pošty pre daného používateľa. Napríklad ak chceme odoslať poшту na adresu *niekto@niekde.sk*, postfix sa spýta služby DNS, kto prijíma poшту pre doménu *niekde.sk* a automaticky tam sám poшту doručí.

Niekedy je však vhodné alebo priam nutné, aby postfix pri odosielaní pošty komunikoval s jedným špecifickým, spravidla nadriadeným SMTP serverom. Vtedy v tejto direktíve dosadíme ako hodnotu adresu nadriadeného servera. Dokonca nepotrebujeme ani poznať meno, stačí jeho IP adresa.

Príklad:

*relay\_host = mail.poskytovatel.sk*

alebo

*relay\_host = 211.181.243.89*

Táto metóda bola zavedená kvôli ochrane pred šírením spamu a vírusov. Niektoré nadriadené SMTP servery si overujú, či sa môžeme na nich pripájať a cez nich poшту odosielať.

Ak je táto hodnota prázdna, server sa pokúsi o doručovanie sám.

**home\_mailbox**

Určuje cestu a spôsob, ako budú správy ukladané pre jednotlivých používateľov. Ak zvolíme hodnotu *Mailbox*, tak bude pošta ukladaná do súboru */var/spool/mail/meno\_používateľa*.

Príklad:

*home\_mailbox = Mailbox*

Toto sú najzákladnejšie nastavenia programu postfix, aby mohol fungovať. My sa k ostatným parametrom ešte vrátíme, keď budeme požadovať špeciálnejšiu funkciu od nášho servera.

Tu uvedené príklady si prispôsobíme k našej sieti a správneho názvu poštového servera.

### Skúška funkčnosti

Aby sme sa presvedčili, že sme SMTP server postfix správne nastavili, vykonáme test odosielania pošty.

Ako prvé sa presvedčíme o syntaktickej bezchybnosti konfiguračného súboru.

Zadáme príkaz *service postfix check*.

Ak dostaneme potvrdenie OK, všetko je, aspoň zo syntaktického hľadiska v poriadku. Prípadné chyby hneď opravíme a server reštartujeme alebo reloadneme.

Teraz však musíme overiť, či náš server je schopný poštu prijať a prípadne odoslať ďalej.

Najprv si overíme, či je schopný prijať poštu pre lokálneho používateľa.

Na linuxe si vytvoríme používateľa s príslušným menom a heslom, v našom prípade používateľa **mior**.

Vieme, že ak na náš server príde pre tohto používateľa pošta, tá sa uloží do súboru */var/spool/mail/mior*.

Pristúpime ku klientskému počítaču, na ktorom chceme, aby si náš používateľ *mior* mohol čítať správy. Teraz nie je dôležité, či na tomto klientovi je nainštalovaný program MUA. Dôležité je, aby bol v dosahu tej istej siete, ako je náš poštový linuxový server FIRMA (to overíme príkazom *ping*).

Je úplne jedno, aký operačný systém má klientsky počítač, z ktorého budeme odosielanie testovať. Podstatné je, aby na ňom fungoval program *telnet*.

A spomeňme si, čo sme sa naučili v predchádzajúcej časti.

Spustíme telnetovú reláciu na porte 25, kde načúva SMTP protokol. V prípade, že nemáme konfigurovanú DNS alebo na klientskej stanici nemáme vytvorený súbor *hosts*, môžeme pre spojenie použiť aj IP adresu nášho linuxového počítača:

```
telnet 192.168.10.1 25
```

alebo

```
telnet mail.firma.sk 25
```

Následnou postupnosťou príkazov vykonáme reláciu (konzola je modrá a odpovede SMTP servera sú červená farba) – výpis č.2:

```
[root@rubin root]# telnet mail.firma.sk 25
Trying 192.168.10.1...
Connected to mail.firma.sk (192.168.10.1).
Escape character is '^]'.
220 mail.firma.sk ESMTP Postfix
helo firma.sk
250 mail.firma.sk
mail from: leonka@doma.sk
250 ok
rcpt to: mior@firma.sk
250 ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: Test poslanca spravy
Toto je skuska poslanca spravy.
.
250 ok: queued as D11E45AF15
quit
221 Bye
Connection closed by foreign host.
```

Meno odosielateľa si môžeme vymyslieť a nekontroluje sa (čo je niekedy kvôli spamom a vírusom škoda...). Veď ani v skutočnej listovej pošte nemusí byť odosielateľ ozajstný!  
Meno príjemcu však musí byť skutočné – v našom prípade to je mior@firma.sk.

Teraz prejdeme k serveru a pozrieme sa do adresára `/var/spool/mail/`. Vidíme, že tam pribudol súbor s menom *mior*. Keď si obsah tohto súboru prezrieme, uvidíme nami odoslanú poštu. Ak otvoríme novú telnetovú reláciu a pošleme ďalšiu správu na tú istú adresu, uvidíme, že nová správa sa pripojila na koniec do už existujúceho súboru *mior*.

Výborne! Takto sme si overili, že sme schopní na náš poštový server správu odoslať. Ten ju prijme a uloží na správne miesto.

Teraz podobným spôsobom odošleme novú správu ďalej, mimo náš poštový server – do sveta. Vieme, že máme zriadený účet mior@poskytovatel.sk. Pošleme správu tam – výpis č.3:

```
[root@rubin root]# telnet mail.firma.sk 25
Trying 192.168.10.1...
Connected to mail.firma.sk (192.168.10.1).
Escape character is '^]'.
220 mail.firma.sk ESMTP Postfix
helo firma.sk
250 mail.firma.sk
mail from: mior@firma.sk
250 ok
rcpt to: mior@poskytovatel.sk
250 ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: Test poslanca spravy
Toto je skuska poslanca spravy.
.
250 ok: queued as D81F46AB11
quit
221 Bye
Connection closed by foreign host.
```

Relácia bude podobná, len v položke **mail from:** môžeme uviesť našu adresu na našom linuxovom serveri, teda mior@firma.sk (ale nemusíme...).

Čo však musíme je, že v položke **rcpt to:** uvedieme skutočnú našu adresu v internetovom svete!

Po ukončení spojenia sa na linuxovom serveri pozrieme do frontu odchádzajúcej pošty príkazom:

```
[root@rubin root]# mailq
```

Ak dostaneme hlásenie **Mail queue is empty**, správa opustila náš poštový server v poriadku.

Ak sa vypíše iné hlásenie, je niekde pri odchode do sveta chyba. Spravidla tu bude aj napísané, prečo správa neodišla.

### **Pozor!**

*Ak posielame dlhú správu, napríklad s prílohou, bude jej odoslanie trvať dlhšie, záleží na rýchlosti pripojenia. Odosielané správy sú označené hviezdikou.*

Teraz sa pomocou vhodného klienta MUA, ktorý máme nakonfigurovaný na poštovú schránku mior@poskytovatel.sk pripojíme k poskytovateľovi a stiahneme si správy.

Ak sa medzi nimi nachádza aj tá naša, ktorú sme odoslali, sme fakt, ale fakt dobrí, lebo sme si overili, že náš poštový server FIRMA dokáže správy odosielať aj do sveta.

Teraz už len zostáva naučiť sa správy zo sveta prijímať. Ale o tom nabadúce.



*Ďalšie veci súvisiace s týmto článkom si môžete stiahnuť z mojej vynovenej web stránky na novej adrese [www.cevaro.sk](http://www.cevaro.sk) a prípadné otázky mi môžete posielat' na adresu [extra@cevaro.sk](mailto:extra@cevaro.sk).  
(Ak vás zaujalo, prečo používam práve doménu CEVARO, tak si toto slovo prečítajte odzadu...)*

# Linux prakticky ako server / 23.časť

V minulej časti sme si nakonfigurovali SMTP server Postfix a odskúšali sme si posielanie správ v rámci našej siete a do Internetu. Zistili sme, že to funguje, čo je prvý predpoklad k tomu, aby sme mohli pokračovať vo výstavbe nášho poštového servera.

Odosielanie pošty však nestačí, treba sa naučiť aj poštu prijímať. A to sa dnes naučíme.

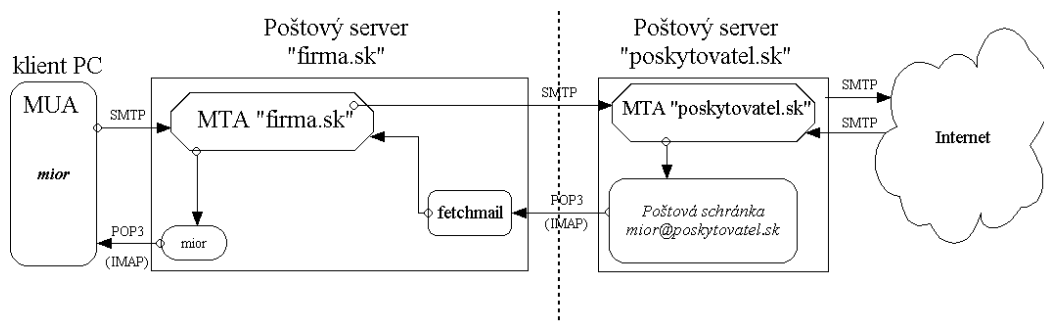
## Popis situácie

Pre lepšie pochopenie dnešnej témy si zopakujme popis situácie:

Všetko nacvičujeme na klasickom poštovom účte u poskytovateľa poštových služieb s menom *poskytovatel.sk*, u ktorého máme vytvorený skutočný účet s názvom účtu *mior*, heslom *sk78cv3* a mejlovou adresou *mior@poskytovatel.sk*.

My vieme, že by sme mohli k tomuto účtu pristupovať aj pomocou MUA a to variantom koncového klienta. Ale aby sme si nacvičili variant s nepriamym doručovaním, vytvoríme si vlastný poštový server *mail.firma.sk*. Na tomto poštovom serveri máme sprevádzkovaný a nakonfigurovaný SMTP server *Postfix* a vytvoreného používateľa *mior*.

Pozrime sa ešte raz na obrázok č.1:



Vidíme, že pošta zaslaná z Internetu na mejlovú adresu *mior@poskytovatel.sk*, sa doručí na poštový server poskytovateľa *poskytovatel.sk* a uloží sa do poštovej schránky s menom identickým s názvom mejlovej adresy. Poštová schránka u poskytovateľa je totiž pre internetovú poštu konečná. A tam zostane, pokiaľ si ju nevyberieme príslušným protokolom POP3 alebo IMAP.

My však takto doručeníu poštu potrebujeme dopraviť ďalej na náš poštový server *firma.sk*! Ako na to?

## Fetchmail

Na výber a prepravu pošty z poštových schránok uložených na iných poštových serveroch použijeme jeden z najrozšírenejších programov s názvom **fetchmail**. Jeho autorom je Eric S. Raymond a nájdeme ho na internetovej stránke <http://www.catb.org/~esr/fetchmail/>.

Ovláda skoro všetky známe protokoly, jednorázové heslá, SSL certifikáty a podobne.

Po nakonfigurovaní a spustení sa program *fetchmail* spojí s požadovanou poštovou schránkou a nájdenú poštu prevezme a odovzdá programu MTA (v našom prípade postfixu – pozri obr.č.1) alebo ju odovzdá inému programu MDA na ďalšie spracovanie (ukážeme si neskôr).

Jeho základné použitie je jednoduché – na príkazovom riadku zadáme používateľa a meno serveru, kde je pošta uložená.

Najzákladnejšie použitie programu *fetchmail* môže byť takéto:

```
[mior@rubin mior]# fetchmail -u mior mail.poskytovatel.sk
```

Po spustení sa program spýta na heslo, nadviaže spojenie so schránkou a pokúsi sa stiahnuť poštu.

Takýto postup je však pre pravidelnejšie sťahovanie pošty nevhodný. Preto existuje možnosť vytvoriť konfiguračný súbor, kde nastavíme požadované parametre. Tento konfiguračný súbor sa nazýva **.fetchmailrc** (aj

s tou bodkou!). Tento súbor uložíme do adresára používateľa, ktorý bude program *fetchmail* spúšťať. Keďže predpokladáme, že budeme *fetchmail* spúšťať s právami *roota*, súbor *fetchmailrc* uložíme do adresára */root/*. V konfiguračnom súbore môžeme definovať aj sťahovanie z viacerých poštových účtov, ktoré môžu byť uložené na rôznych serveroch v Internete.

### Parametre konfiguračného súboru *.fetchmailrc*

Väčšinu vlastností programu *fetchmail* je možné nastaviť aj z príkazovej riadky, ale také isté parametre je možné použiť aj v konfiguračnom súbore.

Teraz si popíšeme najdôležitejšie parametre. Vôbec nie sú všetky, úplný zoznam nájdeme v manuálovej stránke programu *fetchmail*.

Tu je základný zoznam parametrov, ich použitie si ukážeme na konkrétnych príkladoch:

- Ø **-p, --protocol / proto meno\_protokolu** : voľba protokolu. Najčastejšie POP3 alebo IMAP. Ak ne zadáme nič, použije sa AUTO (skúša tri najbežnejšie protokoly IMAP, POP3, POP2)
- Ø **-u, --username / user meno\_úctu** : meno poštového konta. Ak ne zadáme nič, použije sa prihlasovacie meno
- Ø **pass heslo** (iba v konfiguračnom súbore) : heslo poštovej schránky. Ak ne zadáme nič, program sa zakaždým spýta na heslo
- Ø **--auth / auth typ** : spôsob overovania pri prihlásení na server. Ak ne zadáme nič, použije sa password (overovanie heslom)
- Ø **-v, --verbose** (iba v príkazovom riadku): Zapne režim podrobného výpisu. Vhodné pri ladení, ak sa nedarí stiahnuť poštu

### Voľby pre správu správ

Už sme si povedali, že protokol POP3 umožňuje označovať správy ako prečítané. *Fetchmail* implicitne sťahuje iba správy označené ako neprečítané a potom ich zo serveru zmaže. Ku zmene tohto správania slúžia tieto parametre:

- Ø **-k / keep** : ponechať prečítané správy na serveri
- Ø **-K / nokeep** : neponechať prečítané správy na serveri (prebije -k)
- Ø **-F / flush** : pred sťahovaním vymazať všetky správy označené ako prečítané
- Ø **-a / fetchall** : stiahnuť všetky správy, bez ohľadu na to, či sú označené ako prečítané alebo nie.

Mnoho z parametrov nie je potrebné nastavovať, vtedy sa použijú implicitné hodnoty.

### Náš konfiguračný súbor

„Tak ako všetko v *Linuxe*...“ – moja okrídlená veta – tak aj konfiguračný súbor sa dá napísať v rôznych formách. Na výpise č.2 je vzor konfiguračného súboru pre náš variant:

```
poll mail.poskytovatel.sk aka poskytovatel.sk
protocol POP3
user mior password sk78cv3 to mior here

smtpghost rubin.firma.sk
fetchall
```

Kľúčové slovo **poll** hovorí, že bude kontaktovaný daný poštový server, ak nebol na príkazovom riadku žiadny definovaný. Slovo **aka** definuje doménu, ktorú kontaktujeme.

Na druhom riadku definujeme použitý protokol POP3.

Tretí riadok udáva názov a heslo poštového účtu, ktorý chceme vybrať. Slovné spojenie **to mior here** definuje, že vybraná pošta bude uložená do schránky používateľa **mior** na našom (linuxovom) serveri (musí dopredu existovať!).

Parameter **smtpghost** definuje, ktorému serveru bude odovzdaná pošta na ďalšie spracovanie – v tomto prípade to bude odovzdané postfixu na našom serveri.

Parameter **fetchall** vyzdvihne všetku poštu.

### Poznámka:

Neobávajte sa, že sú heslá v otvorenom formáte! Musíme zabezpečiť, aby súbor dostal príslušné práva – teda aby ho mohol čítať a zapisovať iba vlastník (*chmod 600*)!

Predstavme si, že by sme mali ešte jeden účet s menom *miroslav* a heslom *op96kl32* na serveri *mail.cevaro.sk*. Pre jeho sťahovanie nepotrebujeme vytvoriť druhý konfiguračný súbor, ale všetko zapíšeme do tohto jedného súboru. Zároveň si ukážeme druhú formu zápisu – výpis č.3:

```
# spravenie stahovania chceme zaznamenavat v logoch do "/var/log/mail.*"
set syslog

# nastavenie parametrov, ktore su spolocne
defaults      protocol pop3,
               timeout 300,
               nokeep,
               fetchall,

# stiahnut poštu od poskytovatel.sk
poll "mail.poskytovatel.s",
     user "mior"
     password " sk78cv3" is mior here

# stiahnut poštu od cevaro.sk
poll "mail.cevaro.sk",
     user "miroslav",
     password " op96kl32" is mior here
```

Takže stručný súhrn:

Sťahujeme z dvoch účtov. Keďže linka môže byť trochu nekvalitná, tak nastavíme, aby každé spojenie vypršalo po 5 minútach (**timeout 300**). Poštu nebudeme nechávať na serveroch, teda po stiahnutí sa zmaže. Všimnime si, že obidve pošty sa ukladajú do jednej schránky *mior* na našom serveri.

Program *fetchmail* je podstatne flexibilnejší, ako sú uvedené ukážky. Množstvo parametrov dovoľuje rôzne nastavenia a variabilitu.

### Praktický príklad

Teraz si to všetko vyskúšame. Vytvoríme si súbor *fetchmailrc* a uložíme ho do adresára */root/* (je jasné, že premenné parametrov si upravíme podľa svojich možností – svoje meno, svoje heslo atď.). Presvedčíme sa, že na našom linuxovom serveri existuje používateľ *mior*, aby sme mohli do jeho schránky ukladať stiahnutú poštu.

Na našom serveri si overíme, že v adresári */var/mail/* nie je žiadny súbor s názvom *mior*. Ak áno, tak ho vymažeme!

V ľubovolnom klientovi MUA napíšeme správu a odošleme ju do poštovej schránky *mior@poskytovatel.sk*. (Klienta prípadne vypneme, aby nám predčasne sám nestiahol poštu od poskytovateľa - to predsa nechceme, to nech urobí *fetchmail*).

V príkazovom riadku (ako root) zadáme príkaz:

```
[root@rubin root]# fetchmail -v
```

Na obrazovke sa objaví text podobný výpisu č.4:

```
fetchmail: 6.2.0 querying mail.poskytovatel.sk (protocol POP3) at Ne
5. jún 2005, 09:37:56 CEST: poll started
fetchmail: POP3< +OK <99337.1117956164@mail.poskytovatel.sk>
fetchmail: POP3> CAPA
fetchmail: POP3< -ERR authorization first
fetchmail: POP3< +OK <20529.1117956163@mail.poskytovatel.sk>
fetchmail: POP3> USER mior
fetchmail: POP3< +OK
fetchmail: POP3> PASS
fetchmail: POP3< +OK
fetchmail: POP3> STAT
fetchmail: POP3< +OK 1 1933
1 message for mior at mail.poskytovatel.sk (1933 octets).
fetchmail: POP3> LIST
fetchmail: POP3< +OK
fetchmail: POP3< 1 1933
```

```

fetchmail: POP3< .
fetchmail: POP3> RETR 1
fetchmail: POP3< +OK 1933 octets
reading message mior@mail.poskytovatel.sk:1 of 1 (1933 octets)
fetchmail: SMTP< 220 rubin.firma.sk ESMTP Postfix
fetchmail: SMTP> EHLO localhost
fetchmail: SMTP< 250-rubin.firma.sk
fetchmail: SMTP< 250-PIPELINING
fetchmail: SMTP< 250-SIZE 10240000
fetchmail: SMTP< 250-VRFY
fetchmail: SMTP< 250-ETRN
fetchmail: SMTP< 250-XVERP
fetchmail: SMTP< 250 8BITMIME
fetchmail: SMTP> MAIL FROM:<mior@poskytovatel.sk> SIZE=1933
fetchmail: SMTP< 250 Ok
fetchmail: SMTP> RCPT TO:<mior@rubin.firma.sk>
fetchmail: SMTP< 250 Ok
fetchmail: SMTP> DATA
fetchmail: SMTP< 354 End data with <CR><LF>.<CR><LF>
#*****fetchmail: SMTP>. (EOM)
fetchmail: SMTP< 250 Ok: queued as D8C9313C378
not flushed
fetchmail: POP3> QUIT
fetchmail: POP3< +OK
fetchmail: 6.2.0 querying mail.poskytovatel.sk (protocol POP3) at Ne
5. jún 2005, 09:38:00 CEST: poll completed
fetchmail: SMTP> QUIT
fetchmail: SMTP< 221 Bye
fetchmail: normal termination, status 0

```

Však sa tento výpis až náramne podobá na komunikáciu, ktorú sme si skúšali v telnete v minulých častiach!

Na začiatku je spojenie podobné protokolu POP3.

Potom nasleduje riadok „*reading message mior@mail.poskytovatel.sk:1 of 1 (1933 octets)*“, ktorý hovorí, že je prečítaná jedna správa o veľkosti 1933 bajtov.

Potom je vytvorená SMTP komunikácia a riadok „*#\*\*\*\*\*fetchmail: SMTP>. (EOM)*“ symbolizuje, že uvedená správa bola prenesená do nášho linuxového servera (čím je správa dlhšia, tým je počet hviezdíčiek väčší).

Či sa tak naozaj stalo, to sa pozrieme do adresára `/var/mail/`. Tu by sa mal odteraz nachádzať súbor s názvom *mior*.

Tento súbor obsahuje prenesenú správu. Prečítaním obsahu tohto súboru sa presvedčíme, že je to naozaj tak.

Ak pošleme alebo dostaneme ďalšiu správu do poštovej schránky uloženej u poskytovateľa a následne zavoláme *fetchmail* znova, novo stiahnutá správa sa pripojí na koniec súboru `/var/mail/mior`.

### Spúšťanie programu fetchmail

Ako sme si už povedali, program *fetchmail* môžeme spúšťať z príkazového riadku. Toto spustenie je však iba jednorázové. Ak chceme program spúšťať viackrát a v pravidelných intervaloch, musíme ho volať pomocou programu *cron*.

Existuje však ešte jedna – jednoduchšia – možnosť: spustiť program ako démona a stanoviť mu, ako často má kontrolovať poštové schránky.

Na to existuje parameter `-d` (*-daemon*) a číslo, ktoré určuje počet opakovaní v sekundách.

Taktiež môžeme tento parameter zaviesť do súboru `.fetchmailrc` takto:

```
set daemon 300
```

### Poznámka:

Parameter `-d` (*set daemon*) nemôže existovať s parametrom *fetchmail*!

### Upozornenie!

*Nesmieme zabudnúť, že spúšťať program fetchmail ako démona nestačí. Musíme zabezpečiť, aby bol spustený ako démon pri každom štarte systému. Variantov je niekoľko, napríklad cez štartovacie skripty local.rc alebo podobne (toto sme si už vysvetľovali – naštudujte!)*

## Vyberanie pošty z nášho servera

Odtiaľ máme poštu na našom serveri v poštovej schránke *mior*.

No, to že ju tam máme, to je perfektné, ale stále je to pošta na serveri. Aby sme si ju mohli prečítať v ľubovoľnom programe MUA, musíme dovoliť, aby bola pošta z tejto schránky stiahnutá.

My už vieme, že na sťahovanie pošty z poštovej schránky (niekedy aj nazývanej úložisko) do programu MUA používame protokoly POP3 (alebo IMAP).

## Inštalácia a nastavenie protokolu POP3

Aby mohol klientsky program MUA komunikovať s našim linuxovým serverom pomocou protokolu POP3, musíme mať na strane Linuxu nainštalovaného a sprevádzkovaného démona protokolu POP3.

Tento démon je súčasťou každej distribúcie a spravidla býva spojený s balíkom démona IMAP. V distribúcii Red Hat a Fedora je to RPM balíček *imap-xxx.rpm*. Ten obsahuje súbor *imapd*, čo je IMAP démon, súbor *ipop2d*, čo je démon staršieho protokolu POP2 a súbor *ipop3d*, čo je démon POP3, ktorý potrebujeme.

To, či máme v našej distribúcii žiadaný protokol nainštalovaný, si môžeme overiť tak, že skontrolujeme adresár */usr/sbin/*, či sa v ňom nachádza súbor *ipop3d* (alebo podobný).

Tento démon nie je spúšťaný ako služba trvalo, ale pomocou superdémona *xinetd*.

Preto prejdeme do adresára */etc/xinetd.d/* a vyhľadáme súbor *ipop3*.

Jeho obsah je na výpise č.5:

```
# default: off
# description: The POP3 service allows remote users to access their mail \
#               using an POP3 client such as Netscape Communicator, mutt, \
#               or fetchmail.
service pop3
{
    socket_type      = stream
    wait            = no
    user             = root
    server           = /usr/sbin/ipop3d
    log_on_success   += HOST DURATION
    log_on_failure   += HOST
    disable         = no
}
```

Vyhľadáme riadok s voľbou **disable** a nastavíme ju na hodnotu **disable = no**.

Súbor uložíme.

Ak sme previedli zmeny v konfigurácii superdémona *xinetd*, musíme ho reštartovať, aby si načítal nové zmenené hodnoty:

```
[root@rubin root]# service xinetd restart
```

Odtiaľ pri každej požiadavke o komunikáciu s protokolom POP3 bude spustený démon *ipop3d*.

Ak správne nastavíme program MUA, kde zadáme ako poštový server meno alebo IP adresu nášho servera, názov účtu *mior* a heslo, ktoré sme v Linuxe priradili používateľovi *mior*, budeme mať možnosť pripojiť sa z klienta MUA a plnohodnotne komunikovať so serverom – teda poštu nielen odosielať, ale aj vyberať z poštovej schránky na našom serveri firma.sk.

Práve sa nám otvorili dvere do sveta internetovej pošty. Máme funkčný poštový server, vieme odosielať a prijímať poštu. Zatiaľ síce iba zo schránok vytvorených inde, ale predsa. Niektorí namietne, že sa to dá celé obísť koncovým klientom!

Áno, ale my si odtiaľ môžeme s došlou (aj odosielanou) poštou robiť, čo chceme! Môžeme ju filtrovať, upravovať, kopírovať, archivovať.

Ale hlavne na základe tohto príkladu budeme schopní vybudovať server, ktorý bude sťahovať poštu pre celú firmu či školu – a to z doménového koša.

A to si ukážeme nabudúce!

*Uvedené programy, ako aj ďalšie iné veci súvisiace s týmto článkom, si môžete stiahnuť z mojej vynovenej web stránky na novej adrese [www.cevaro.sk](http://www.cevaro.sk), sekcia Download, a prípadné otázky mi môžete posilať na adresu [linux@cevaro.sk](mailto:linux@cevaro.sk).*

*Miroslav Oravec*