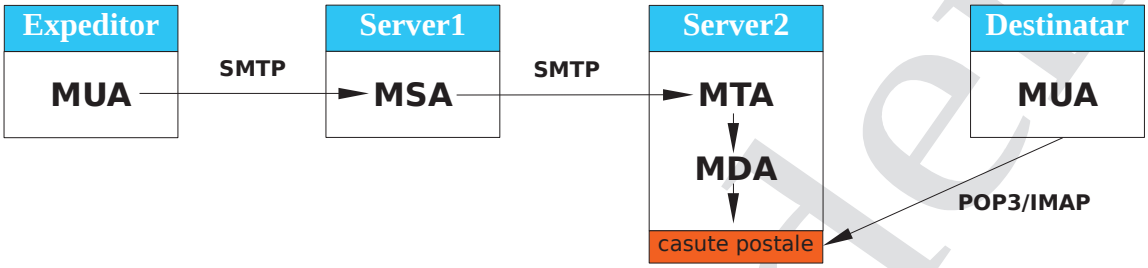
12.SERVERUL E-MAIL

12.1. Sistemul de posta electronica

12.1.1. Lantul de transmitere a e-mail-ului in internet

Ca si in cazul sistemului postal obisnuit, un mesaj e-mail trece prin mai multe „maini” inainte de a ajunge la destinatar. Atunci cand dorim sa trimitem o scrisoare, o scriem si apoi o depunem la cea mai apropiata cutie postala; de acolo ea este transportata la cel mai apropiat oficiu postal, unde, in functie de adresa destinatarului, fie ii va fi livrata direct acestuia prin intermediul postasului (daca destinatarul se afla in zona acoperita de acel oficiu), fie va fi transmisa mai departe oficiului postal in aria caruia se gaseste destinatarul. De aici scrisoarea va fi preluata de catre postas si depusa in cutia destinatarului, unde va zabovi pana cand acesta o va ridica.

Lantul de transmisiune al unui e-mail in internet poate fi gandit prin analogie:



Scrierea scrisorii	Expeditorul editeaza mesajul (folosind Thunderbird, Outlook Express etc), folosind ca adresa destinatie contact@infoacademy.net
Depunerea la cutia postala	Expeditorul apasa „Send” in clientul de mail
Scrisoarea este preluata si dusa la oficiul postal	Clientul de mail livreaza mesajul catre „outgoing server” specificat la configurarea contului; acesta este Server1 din figura de mai sus
Este destinatarul in aria de acoperire a oficiului?	Este Server1 configurat ca destinatie finala pentru domeniul DNS infoacademy.net (partea dreapta a adresei)? In cazul nostru nu
Scrisoarea este livrata oficiului postal de care apartine destinatarul	Server1 determina cine este serverul destinatie legitim al mesajului, gasindu-l in “persoana” lui Server2. Server1 se conecteaza la Server2 si ii livreaza mesajul. Server2 il va accepta deoarece este configurat ca destinatie finala pentru domeniul infoacademy.net
Factorul postal duce scrisoarea la cutia destinatarului	Serverul destinatie livreaza mesajul in casuta postala (mailbox) a destinatarului. Casuta consta dintr-unul sau mai multe fisiere sau/si directoare din sistemul de fisiere al serverului
Destinatarul isi ridica scrisoarea	Destinatarul se conecteaza la Server2, isi acceseaza casuta si vizualizeaza mesajul

Figura de mai sus prezinta cazul cel mai des intalnit in transmiterea unui mesaj email. Istoric vorbind insa, sistemul nu a functionat intotdeauna asa; iata in cateva paragrafe evolutia sa si ratiunile care l-au adus la forma curenta:

- in faza incipienta, cand inca nu existau retele, mesajele reprezentau o forma de comunicare locala, intre utilizatorii aceluiasi sistem (dam ca exemplu statiile mainframe, pe care puteau lucra mai multi useri folosind terminale fizice)
- odata cu aparitia retelelor, sistemul a fost extins astfel incat mesajele sa poata fi directionate catre useri ai altor statii; mailul avea ca expeditor un user de sistem de pe o statie, iar destinatarul era tot un user de sistem dar al altei statii. Pratic utilizatorul logat cu *user1* pe 10.0.0.1 trimitea mail catre *user2@10.0.0.2*. Aceasta presupunea faptul ca statia destinatie rula un soft de server care accepta mesajele de la clientii din retea (element nou fata de cazul anterior). Asadar, in acest stadiu, figura de mai sus nu ar fi continut *Server1* si *Server2*, ci doar masina expeditorului si cea a destinatarului
- odata cu interconectarea retelelor si aparitia internetului, solutia anterioara nu mai era satisfacatoare. Destinatarul uman putea lucra pe diferite statii (si mai nou dispozitive) din retea sau din internet, avand insa in continuare nevoie de acces la mailul propriu. Era nepractic si costisitor sa se creeze un sistem care sa incerce livrarea direct catre masina curenta a destinatarului, oricare ar fi ea. De aceea sistemul a fost ajustat prin introducerea lui *Server2*, care reprezinta locul stabil al casutelor postale pentru un intreg set de destinatari, si care este o masina usor determinabila prin intermediul DNS; odata mesajele livrate in casuta, destinatarul poate accesa de la distanta casuta personala. In acest scop au fost create seturi de protocoale separate: unul pentru portiunea

expeditor-casuta si altul pentru segmentul casuta-destinatar. Conform acestei logici, lantul de transmisiune ar avea pana in acest moment 3 statii: expeditor, *Server2* si destinatar

- s-a constatat ca nu este practic ca expeditorul sa se conecteze direct la *Server2*, desi acest lucru este tehnic posibil. *Server1* a fost introdus in acest lant de transmisiune deoarece ofera cateva avantaje:
  - securitate
    - *Server1* poate verifica identitatea clientilor sai, netrimitand mesaje decat in numele celor autentificati corect
    - *Server1* poate reprezenta punctul unic de iesire din retea a mesajelor e-mail, ceea ce permite configurarea unor firewall-uri cu reguli mai drastice si o mai buna protectie impotriva spam-ului. Sa ne gandim la faptul ca multi virusi ai zilelor noastre au ca efect trimiterea de mailuri nesolicitate; daca o retea ar permite accesul direct al tuturor statiilor sale la servere e-mail externe, ar fi complicat de contracarat efectul unui astfel de virus
  - o mai buna gestionare a erorilor. Spre exemplu, atunci cand expeditorul nu ar putea livra un mesaj catre *Server2* din cauza unei probleme temporare (una de conectivitate sau o eroare a lui *Server2*), sarcina retry-urilor i-ar reveni expeditorului. Nici in trecut, nici in ziua de astazi, expeditorul nu reprezinta neaparat o statie fixa si cu conexiune permanenta la internet; in trecut se conecta folosind un modem analogic, iar in ziua de astazi poate fi un dispozitiv mobil a carui conexiune nu este permanent activa. In aceste conditii nu este convenabil ca eventualele retry-uri sa fie facute de catre statia/dispozitivul expeditorului; in momentul in care expeditorul are conectivitate, mesajul ii este livrat lui *Server1*, care este o masina cu adresa si conexiune internet stabile, permanent pornita, si care este deci mai potrivita pentru comunicatia directa cu *Server2*, oricare ar fi meandrele acestui dialog. *Server2* preia responsabilitatea procesarii mesajelor si a eventualelor retry-uri. In cazul unei erori emise de *Server2*, expeditorul va fi instiintat la urmatoarea sa conectare la *Server1*

In sistemul ilustrat mai sus intervin doua mari categorii de elemente:

- **aplicatii software: MUA, MSA, MTA, MDA**. Sunt cele care creeaza sau proceseaza mesajele email. Acestea nu reprezinta denumiri de softuri, ci roluri indeplinite in cadrul sistemului; pentru fiecare dintre cele 4 roluri exista o multitudine de solutii software disponibile
- **protocoale: SMTP, POP, IMAP**. Sunt cele prin intermediul carora se transfera mesajele de la o statie la alta

Ambele categorii vor fi detaliate in cele ce urmeaza.

12.1.2. Componente software ale sistemului e-mail

Sistemul email reprezinta un ansamblu de software-uri care conlucreaza, fiecare avand un rol bine stabilit:

1. **MUA – Mail User Agent**. Reprezinta programul folosit de utilizator pentru a compune, trimite si primi mailul (ex: Outlook, Thunderbird, Evolution etc.)
2. **MSA - Mail Submission Agent**. Este softul care accepta mesaje de la MUA-uri autentificate, le aplica eventuale filtre/corectii si le livreaza unui MTA
3. **MTA – Mail Transfer Agent** (numit si *mail relay*) - reprezinta softul care joaca rolul de intermediar in pasarea mesajelor intre doua statii. El poate indeplini atat rol de server (acceptand mesaje de la clienti) cat si de client (atunci cand livreaza mesaje catre alte servere). Lantul de transmisiune a unui mesaj poate include mai multe MTA-uri succesive
4. **MDA – Mail Delivery Agent** - reprezinta softul ce joaca rolul de dispecer pentru mesajele primite de la MTA, fie plasand mesajele direct in casutele postale corespunzatoare (local delivery), fie trimitandu-le la alta adresa (forwarding) sau inmanandu-le spre prelucrare unui alt program (de exemplu, un analizor de spam sau un antivirus). MDA este in general un modul al softului de server SMTP, insa exista si softuri de tip MDA de sine statatoare

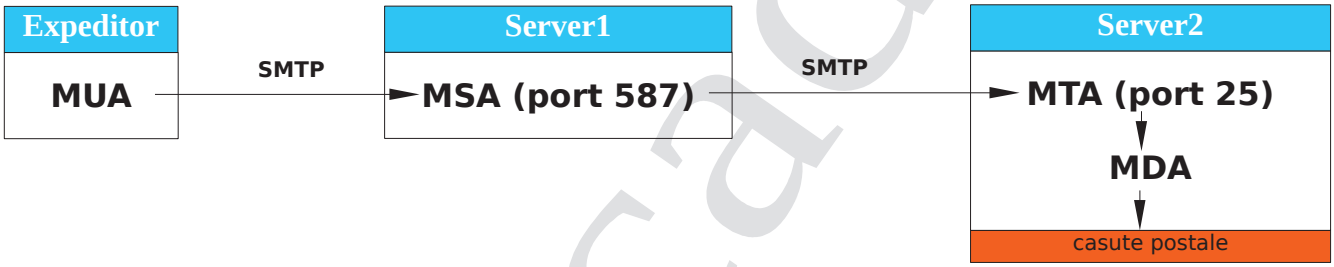
Cu exceptia MUA, softurile prezentate mai sus se gasesc in general in compozitia oricarui server SMTP: acesta poate indeplini rolul de MTA sau MSA (sau ambele!) si contine in general unul sau chiar mai multe module de tip MDA. Exista insa si softuri separate ce indeplinesc cate unul singur dintre rolurile amintite mai sus (ex: *procmail* sau *maildrop*, care joaca rolul de MDA)

Initial, sistemul nu continea conceptul de MSA; serverele SMTP (numite initial, nediferentiat, MTA-uri) erau cele care indeplineau rolul de receptie a mesajelor de la clientii lor, indiferent daca acestia erau MUA-uri sau MTA-uri. Serverul asculta pe portul 25 in toate cazurile. Cu timpul insa au aparut aspecte ce solicitau o separare de roluri:

- in ziua de astazi, in cele mai multe cazuri clientii finali (MUA-urile) trebuie sa se autentifice pentru a putea trimite mail printr-un server SMTP (segmentul expeditor-Server1 din lantul de transmisiune). In schimb, in relatia MTA-MTA (segmentul *Server1-Server2* din figura) deseori nu exista autentificare, deoarece cele doua servere nu au o relatie anterioara; *Server2* poate primi mesaje de la orice alt server din internet, cu care nu are o relatie preexistenta
- unele MUA-uri nu se conformau standardelor si trimiteau mesaje incorecte sau incomplete. In consecinta, MTA-ul ce primea mesajul trebuia sa efectueze corectiile necesare. Problema este ca era dificil de facut distinctia corecta intre clientii de tip MUA si alte MTA-uri, astfel incat sa fie rescrise numai mesajele primite de la clientii finali
- atunci cand administratorul unei retele dorea ca clientii sai sa poata folosi serverul SMTP al retelei pentru a trimite mail cand se afla afara retelei, trebuia sa lase deschis portul 25 in firewall; din nefericire acelasi port era cel folosit pentru mesajele provenite de la alte servere si destinate utilizatorilor din retea, ceea ce facea mai dificila separarea celor doua categorii de trafic si combaterea spam-ului

Introducerea MSA (vezi RFC 6409) a insemnat separarea clara a operatiei de remitere (“submission”) de cea de livrare/transfer. In consecinta, in ziua de astazi exista urmatoarele diferente intre MSA si MTA:

- MSA este obligat sa-si autentifice clientii, pe cand MTA nu
- MSA poate aduce modificari mesajelor, pe cand in cazul MTA acest lucru este in general contraindicat
- MSA functioneaza pe portul 587, pe cand MTA pastreaza portul original, 25



12.1.3. Protocoale componente si roluri

Comunicatia intre diversele statii din lantul de transmisiune e-mail prezentat la inceput este asigurata de urmatoarele protocoale:

- **SMTP** (Simple Mail Transfer Protocol - RFC 5321) - este protocolul ce are ca responsabilitate transmiterea mesajului pana pe serverul ce contine casuta destinatarului. El este folosit atat pe segmentul MUA-MSA cat si MTA-MTA; diferenta este ca, in primul caz, serverul asculta pe portul 587 si solicita autentificare, pe cand in cel de-al doilea serverul asculta pe portul 25 si autentificarea nu este obligatorie
- **POP3** si **IMAP** - sunt protocoalele folosite de catre destinatar pentru accesarea de la distanta a propriei casute postale. Ambele protocoale ii solicita clientului sa se autentifice

Trebuie inteles faptul ca trimiterea unui mesaj incepe din MUA-ul expeditorului si se incheie in casuta postala a destinatarului (asadar nu neaparat pe masina/dispozitivul destinatarului!); in figura noastra, mailul se considera transmis cu succes odata ce a ajuns in casuta postala corecta de pe *Server2*, moment in care responsabilitatea protocolului SMTP se incheie. In continuare destinatarul este cel care initiaza accesarea de la distanta a casutei sale prin intermediul protocoalelor POP3 sau IMAP. Putem asadar gandi diagrama ca fiind impartita in doua zone distincte – cea corespunzatoare *transmiterii* e-mail-ului, si cea legata de *extragerea* acestuia de la distanta – care au in comun casuta postala de pe serverul 2: softul de server SMTP depune mesaje in casuta, iar softul de POP/IMAP le extrage/manipuleaza la cererea MUA al destinatarului.

Remarcam ca, atunci cand vrem sa facem un server de mail care sa receptioneze mesaje pentru un domeniu si de pe care utilizatorii sa le poata descarca/vizualiza, sunt necesare cel putin doua softuri:

- server SMTP, pentru ca utilizatorii sa poata primi mail
- server POP sau IMAP (sau ambele), pentru ca utilizatorii sa-si poata accesa casutele postale.

12.1.4. Livrarea email-ului

12.1.4.1. Relatia cu DNS

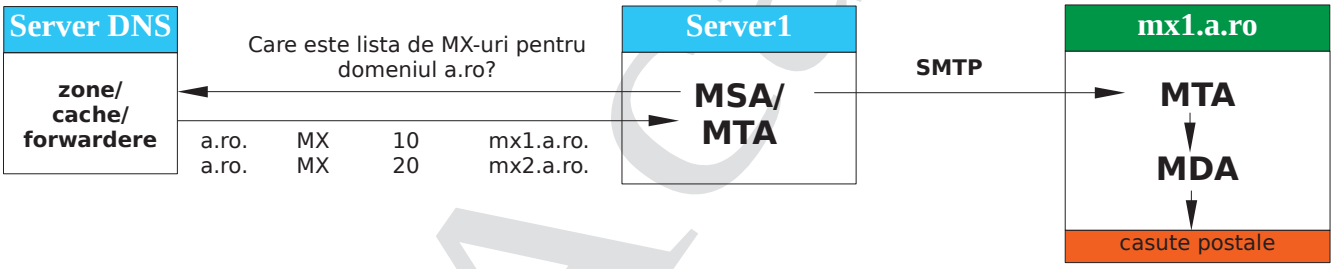
Un server SMTP intermediar (MSA sau MTA) care a receptionat un mail de la un client si care trebuie sa-l livreze mai departe catre serverul destinatie dispune de o singura informatie pe care o poate folosi pentru a trimite mesajul catre serverul corect: adresa email destinatie. El nu cunoaste serverul pe care se afla casuta destinatarului, insa ii poate afla adresa folosindu-se de informatii prezente in sistemul DNS.

Partea dreapta a adresei e-mail (contact@**a.ro**) reprezinta domeniul DNS destinatie. Daca dorim sa putem receptiona mail pentru domeniul nostru, trebuie sa adaugam in fisierul zona DNS inregistrari de tip MX corespunzatoare acestui nume si inregistrari de tip A asociate, ca in exemplul urmator:

```
a.ro.          SOA     ns1.a.ro.     admin.a.ro.   13     3600   7200   14400  60
               NS      ns1.a.ro.
ns1.a.ro.      A       1.2.3.4
mx1.a.ro.      A       5.6.7.8
mx2.a.ro.      A       9.0.1.2
a.ro.          MX      10     mx1.a.ro.     ; MX pointeaza catre un nume de statie, nu catre un IP!
a.ro.          MX      20     mx2.a.ro.
```

Cel de-al treilea camp al inregistrarii MX reprezinta prioritatea, care este cu atat mai buna cu cat numarul este mai mic.

Atunci cand un MTA incearca sa determine serverul SMTP destinatie al unui mesaj, el ii cere resolver-ului DNS lista de RR-uri de tip MX pentru domeniul tinta (partea dreapta a adresei email) si le ordoneaza dupa prioritate, incercand sa trimita catre serverul cu prioritatea cea mai buna. Daca acesta nu poate fi contactat se incearca serverul de pe pozitia urmatoare din lista de prioritati s.a.m.d.



Inregistrarile MX multiple pot fi folosite in diferite scopuri:

- **load balancing**. In cazul unui flux mare de mesaje, se pot configura mai multe servere cu prioritatea optima (numeric minima), astfel incat incarcarea sa se distribuie intre ele. In general acestea vor livra mesajele intr-un spatiu de stocare comun, altminteri mesajele utilizatorilor s-ar imprastia pe mai multe servere
- **fail-over**. In cazul in care serverul principal (cel cu prioritatea optima) este inaccesibil, clientii vor incerca sa le livreze mesajele serverelor cu prioritate inferioara. Un astfel de server “secundar” poate fi configurat fie sa livreze mesajul la casuta (in cazul in care spatiul de stocate al casutelor este comun cu serverul primar) fie sa accepte mesajul dar sa i-l livreze primarului atunci cand acesta revine online. Serverele secundare care actioneaza in acest fel sunt numite si “backup MX”

Atentie! Numele statiilor desemnate ca MX trebuie sa se regaseasca in fisierul zona sub forma de inregistrari de tip A! Daca se folosesc CNAME-uri este posibil ca livrarea sa nu functioneze.

```
mx1.a.ro.      A              1.2.3.4
mail.a.ro.     CNAME          mx1.a.ro.
~~a.ro.~~      ~~MX~~     ~~10~~     ~~mail.a.ro.~~ ;nu e recomandabil ca partea dreapta a MX-ului sa fie CNAME!
a.ro.          MX     10     mx1.a.ro.  ;varianta corecta
```

12.1.4.2. Criterii de acceptare a unui mesaj. Rolul de relay

Odata ce un server SMTP a acceptat un mesaj, inseamna ca isi asuma responsabilitatea sa il proceseze in numele clientului emitent, orice ar insemna aceasta. Decizia de acceptare a mesajelor poate fi luata in functie de orice caracteristica a clientului sau mesajului in cauza; in general insa sunt folosite doua mari strategii de baza, in functie de rolul serverului:

- in cazul unui MSA, decizia acceptarii mesajului se ia in functie de caracteristici ale clientului (ex: adresa IP) sau continut al mesajului (ex: i se pot aplica acestuia filtre antivirus/antispam) dar, mai ales, in functie de identitatea clientului obtinuta in urma autentificarii acestuia. Odata mesajul acceptat, el este transmis mai departe catre un MTA
- in cazul unui MTA, serverul va accepta mesajul daca domeniul DNS destinatie (partea dreapta a adresei email a destinatarului) se regaseste in lista de domenii pentru care serverul a fost configurat sa accepte mailuri. Exista aici doua scenarii distincte:
  - serverul reprezinta destinatia finala a mesajului si deci va incerca livrarea lui la casuta prin intermediul MDA-ului. Acesta are la randul sau de luat niste decizii (vezi 12.1.4.3)
  - serverul este un backup MX, acceptand mesajul in mod artificial (caci nu el este destinatia finala). In aceasta situatie serverul va folosi modulul sau de client SMTP pentru a-i transmite mesajul serverului cu prioritate optima al domeniului destinatie

Doua dintre cazurile descrise mai sus (MSA, respectiv MTA cu rol de backup MX) prezinta interes suplimentar. Atunci cand un server accepta un mesaj desi el nu reprezinta destinatia de drept a acelui mesaj (nu este cel care contine casuta destinatie), spunem ca joaca rolul de **relay**. Un server care accepta mesaje de la orice client si le livreaza catre orice server destinatie poarta numele de **open relay;** trebuie evitata configurarea unui server in acest mod, deoarece el poate fi folosit de catre infractori ca intermediar in trimiterea de spam.

Conform explicatiilor anterioare, constatam urmatoarele:

- MSA joaca un rol de relay selectiv - el va livra mailuri catre alte destinatii insa numai pentru un set bine delimitat de clienti
- MTA joaca rolul de relay doar in cazuri speciale (ex: backup MX, gateway e-mail etc), deoarece el reprezinta deseori destinatia finala a mesajelor

Nota: *in continuare vom numi domenii locale ("local domains") ale unui server lista de domenii ale caror casute se gasesc pe serverul in cauza; cu alte cuvinte, domeniile pentru care serverul accepta mesaje deoarece se stie responsabil pentru ele, fara a juca rolul de relay.*

12.1.4.3. Procesarea mesajelor de catre MDA

12.1.4.3.1. Actiuni posibile aplicate unui mesaj

Atunci cand mesajul este destinat unuia dintre domeniile locale, MDA-ul trebuie sa determine in ce fel trebuie el procesat. Altfel spus, sa faca corespondenta email ↔ actiune. Din acest punct de vedere, in spatele unei adrese email se poate afla:

- **o casuta postala locala a unui utilizator**. In acest caz, MDA va efectua livrarea mesajului in casuta ce corespunde adresei
- **un alias**. Un alias reprezinta o adresa virtuala, care se poate expanda intr-una sau mai multe actiuni, cum ar fi:
  - ***un nume secundar dat unei casute postale existente***. Spre exemplu, sa presupunem ca adresa *contact@firma.ro* exista si are o casuta corespondenta, si dorim ca mailurile destinate adresei *office@firma.ro* sa fie livrate in aceeasi casuta. Vom defini un alias *office* ce trimite catre casuta locala existenta *contact*. De remarcat ca, in acest caz, alias-ul este o solutie locala, un simplu nume secundar de casuta, ceea ce il face asemanator cu conceptul de hard link multiplu: asa cum un hard link nu putea functiona inter-partitii, alias-urile de acest fel nu functioneaza inter-servere
  - ***un forward (redirectionare)***. Mesajul este trimis de aceasta data catre o alta adresa – asadar tinta nu mai este un simplu nume de casuta, ci o adresa completa. Adresa se poate afla pe acelasi server sau pe unul extern. Situatia este asemanatoare cu cea a symlink-ului din sistemul de fisiere
- **nimic**. Atunci cand MDA-ul determina ca nu exista nicio casuta sau actiune corespunzatoare adresei solicitate, produce un mesaj de eroare ce poarta denumirea de *bounce*. Mesajul in cauza are in general drept expeditor

contul MAILER-DAEMON si contine informatii utile precum data/ora bounce-ului, serverul care l-a produs, motivul erorii si headerele sau chiar continutul (partial sau complet) al mesajului initial

Un alias poate avea drept corespondent o intreaga lista de actiuni de naturi diferite; in cazul maximal, un mesaj destinat unei astfel de adrese virtuale poate fi livrat local catre casuta originala, catre una sau mai multe casute suplimentare si, in plus, redirectionat catre una sau mai multe adrese.

Data fiind flexibilitatea alias-urilor, ele pot fi folosite pentru a implementa felurite facilitati, printre care se numara:

- adrese de grup. Atunci cand este nevoie sa se trimita mail in mod repetat catre acelasi grup de adrese, se poate defini un alias care are drept corespondent lista de adrese email ce formeaza grupul (ex: alias-ul *conducere@firma.ro* poate trimite mesajele catre *sef1@firma.ro* si *sef2@firma.ro*)
- pentru a directiona catre aceeasi casuta adrese standard ale unui domeniu. Standardele ce guverneaza sistemul de mail impun ca fiecare domeniu ca aiba definite cateva adrese standard (*postmaster@domeniu*, *abuse@domeniu* etc). Deseori nu este convenabil ca acestea sa fie casute separate; mai degraba ele sunt alias-uri catre casutele sau adresele persoanelor ce au competentele aferente

12.1.4.3.2. Conturi si casute

Fiecare casuta postala corespunde unui anumit utilizator. Conturile de utilizator pot fi:

- **de sistem** – utilizatorul are o identitate definita in */etc/passwd*.
- **virtuale** – utilizatorul este definit intr-o baza de date separata fata de cea de sistem si utilizata numai de catre serverul SMTP si eventual POP3/IMAP

Lucrul cu useri virtuali reprezinta un subiect avansat si care depaseste cadrul acestui material; vom mentiona doar cateva aspecte comparative ale lucrului cu cele doua tipuri de useri, pentru o mai buna informare a cititorului:

- useri de sistem
  - avantaj: usurinta de administrare. Exista utilitare binecunoscute pentru administrarea de conturi de sistem, iar fiecare user de sistem devine automat adresa de mail (ex: daca cream userul *office*, iar serverul email este configurat sa receptioneze mail pentru domeniul *a.ro*, automat a devenit disponibila adresa *office@a.ro*)
  - dezavantaje:
    - fisierul */etc/passwd* are un set prestabilit de campuri, potrivite pentru serviciul de login. Nu putem adauga campuri suplimentare, desi in lumea email ele sunt deseori necesare/practice (ex: quota, acces POP/IMAP etc)
    - posibile conflicte de nume atunci cand serverul gazduieste mai multe domenii care au nevoie de adrese cu acelasi username (ex: dorim ca serverul nostru sa receptioneze mail pentru domeniile *a.ro* si *b.ro*, existand adresele office@a.ro si office@b.ro; nu avem cum crea doi useri de sistem numiti *office*)
    - schimbarile efectuate in */etc/passwd* au nevoie de privilegii de root
- useri virtuali
  - avantaj: flexibilitate. Userii pot fi memorati in baze de date alternative (ex: server SQL), eliminand limitarile lui *passwd* si nemainecesitand privilegii de root
  - dezavantaj: configurarea este mai complicata, solicitand competente sporite

Trebuie inteles faptul ca, atunci cand vine vorba de livrarea unui mesaj la casuta, serverul SMTP foloseste baze de date cu utilizatori numai pentru a stabili corespondenta adresa e-mail → casuta postala; el nu are nevoie de parola utilizatorului decat daca a fost compilat cu extensii de autentificare SMTP (vezi SASL2), in situatia in care serverul joaca rolul de MSA. Asadar, pentru a crea un server de tip MTA care foloseste numai conturi de sistem, utilizatorii definiti nu este nevoie sa aiba si drept de login in sistem – ba chiar este contraindicat.

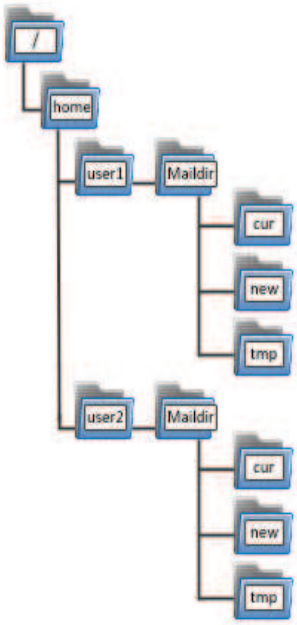
12.1.4.3.3. Formate de casuta

O casuta postala (mailbox) este locul in care se stocheaza mesajele destinate unui anumit utilizator. Ea poate lua diferite forme – un fisier, mai multe fisiere, un director, sau chiar informatie stocata pe un server de baze de date.

Istoric vorbind, formatul de casuta a evoluat si el odata cu restul sistemului e-mail. Exista aceste mari abordari ale stocarii mesajelor email:

- **mbox**. Aceasta este denumirea generica data solutiilor care stocheaza casuta intr-un singur fisier text. Fiecare mesaj nou este adaugat ca atare la sfarsitul fisierului. Solutia are performante bune cand vine vorba de accesarea/cautarea mesajelor, insa prezinta diferite dezavantaje, dintre care amintim:
  - formatul text este ineficient pentru stocare si manipulare de informatii de anvergura (a se considera cazul unui server IMAP, unde mesajele se pot acumula in cantitati mari)
  - in cazul mailului accesat din retea (cazul zilelor noastre) sunt necesare mecanisme de locking (incuiere) a casutei pentru a preintampina problemele legate de accesul concurent la casuta. Spre exemplu, daca un client incearca sa-si acceseze casuta prin POP3 in timp ce serverul SMTP livreaza un mesaj, este posibil ca userul sa vada un mesaj partial sau chiar sa primeasca o eroare caci casuta este perceputa ca fiind corupta
- **maildir**. Formatul maildir a fost introdus pentru a elimina problemele legate de locking. Casuta ia forma unui director cu 3 subdirectoare distinctive: *new* (mesaje noi), *cur* (mesaje curente) si *tmp* (spatiu temporar de schimb intre primele doua directoare). Fiecare mesaj este stocat intr-un fisier propriu cu nume unic, intr-unul dintre primele doua directoare. Mesajele noi livrate in casuta sunt plasate in *new*, iar in momentul in care MUA-ul afiseaza lista de mesaje sunt mutate in directorul de mesaje curente. Astfel se realizeaza separarea intre servere: serverul SMTP livreaza mesajele in primul director, iar serverele POP/IMAP vor opera pe mesajele din cel de-al doilea director, fara a interfera cu primul

  Nota: *numele directorului ce constituie casuta postala nu este impus, el putand fi stabilit de catre administrator din setarile serverului.*



Casute postale in format maildir

- **formate proprietare** – exista o multitudine de softuri de tip server SMTP care au introdus propriul format pentru stocarea mesajelor, ce aduce un plus de eficienta sau/si securitate comparativ cu formatele clasice. Acest lucru insa exclude interoperabilitatea cu alte servere

12.1.5. Accesarea emailului de catre destinatar

12.1.5.1. Scenarii si solutii

Odata mesajul ajuns in casuta postala de pe serverul destinatie, responsabilitatea sistemului de livrare a mailului se incheie. De aici incolo destinatarul este cel care trebuie sa-si acceseze casuta, existand in acest scop diferite posibilitati:

- acces direct la casutele postale
  - **local** – destinatarul este un user local al serverului destinatie, conditii in care MUA se afla pe aceeasi statie ca si casuta si deci o poate accesa direct. Solutia este posibila de la origini si pana in ziua de astazi, dar nu este foarte practica deoarece presupune login-ul pe serverul destinatie, ceea ce nu se intampla pentru majoritatea beneficiarilor uzuali ai sistemului de posta electronica
  - **de la distanta** – MUA se afla pe alta statie insa are acces la un share in care se gasesc mailbox-urile. Este o solutie nesigura, problematica si, in consecinta, de evitat
- acces indirect la casutele postale, prin protocoale special gandite in acest scop:
  - **POP3** (Post Office Protocol) – un protocol simplu, cu care mesajele sunt in general downloadate de catre client si sterse de pe server
  - **IMAP** (Internet Message Access Protocol) – un protocol mai complex care permite mentinerea si structurarea mesajelor pe server

12.1.5.2. Protocolul POP3

Post Office Protocol a fost gandit pentru a le oferi utilizatorilor posibilitatea de a-si descarca de la distanta mesajele aflate pe un server e-mail prin intermediul unui protocol simplu. Protocolul porneste de la premiza ca, in cele mai multe cazuri, mailul este downloadat pe statia destinatarului, si de aceea nu ofera modalitati de structurare a informatiei pe server sau

de control granular al informatiilor traficate intre client si server; exista posibilitatea descarcarii partiale a mesajelor, dar implementarea este una simplista.

Utilizarea acestui protocol presupune prezenta unui soft de server POP care sa aiba acces la casutele postale; in cele mai dese cazuri acesta ruleaza chiar pe statia ce depoziteaza casutele. Serverul POP asculta pe portul TCP 110. Fiecare client este autentificat dupa conectare; in cazul unei autentificari reusite, clientul primeste acces la casuta sa postala si poate descarca/sterge mesaje.

Odata descarcat un mesaj si sters de pe server, el nu mai poate fi accesat decat pe masina pe care se afla. Acest lucru nu reprezenta un inconvenient in trecut, cand userii erau mult mai legati de statii si mobilitatea era redusa; in ziua de astazi insa este omniprezenta necesitatea de a avea acces la mailurile proprii din diverse locatii si de pe felurite device-uri. In consecinta, a fost creat din timp un alt protocol, capabil sa satisfaca aceste noi nevoi.

12.1.5.3. Protocolul IMAP

Protocolul IMAP porneste de la premiza ca mesajele raman stocate pe termen lung in casuta postala, iar aceasta este accesata din locatii si cu MUA-uri multiple. In consecinta, protocolul ofera servicii suplimentare fata de POP3:

- **sincronizare controlata**. Mesajele stau pe server, iar clientii nu fac altceva decat sa se sincronizeze cu serverul; clientul trebuie sa aiba controlul asupra listei de mesaje descarcate si a cantitatii de informatie/timpului pe care le implica sincronizarea (sa ne imaginam ce ar insemna sa ne configuram pe un dispozitiv mobil (smartphone/tableta/etc) un cont IMAP in al carui inbox se afla deja 10000 de mesaje, daca ar trebui sa le descarcam integral!). In acest scop, protocolul permite un control foarte granular al informatiei care circula intre server si client
- **structurarea informatiei pe server**. Informatia acumulandu-se permanent pe server, este incomod ca toate mesajele sa stea la gramada intr-o singura casuta; userul are in general nevoie sa le structureze, sa le categoriseasca. IMAP ofera posibilitatea definirii de casute secundare care actioneaza din punct de vedere al utilizatorului ca niste foldere, permitand gruparea avantajoasa a mesajelor

In timp, mesajele se acumuleaza pe server; implicit este necesar un provisioning mai atent. Statia pe care se afla casutele postale trebuie sa aloce mai mult spatiu pentru acestea comparativ cu un server POP, si deseori sunt necesare mecanisme de limitare a spatiului ocupat per casuta (quota).

Serverul IMAP asculta pe portul 143 TCP.

12.2. Serverul SMTP postfix

12.2.1. Descriere generala

Postfix este un server SMTP sigur, usor de folosit si gandit sa inlocuiasca clasicul sendmail – serverul traditional Unix/Linux, care la vremea dezvoltarii postfix-ului avea in spate o intreaga istorie de probleme de securitate. Postfix este modular – serverul este compus din numeroase utilitare care lucreaza fiecare cu privilegiile minim necesare pentru a-si duce la indeplinire indatoririle.

Serverul postfix poate indeplini diferite roluri din sistemul email:

- poate functiona ca server SMTP primar sau de backup pentru un domeniu
- poate receptiona mail pentru unul sau mai multe domenii, folosind useri de sistem sau virtuali
- permite functionarea selectiva ca intermediar (relay) in functie de parametrii clientului sau de continutul mesajului
- suporta autentificare SMTP prin intermediul SASL (Simple Authentication and Security Layer – RFC 4422,4954) ceea ce ii permite functionarea ca MSA

12.2.2. Fisa serverului

Iata principalele informatii necesare operarii si diagnosticarii serverului:

- comenzi:
  - **postfix** – folosită pentru pornirea/oprirea serverului și validarea/recitirea configurarii
  - **postconf** – utilizata pentru manipularea configurarii serverului
  - **postqueue** – comanda folosita pentru managementul cozii de mesaje a serverului
- pornire: **postfix start**. Are ca efect aparitia a 3 procese: *master*, *qmgr* si *pickup*. Procesul *master* este coordonatorul tuturor proceselor postfix; el ruleaza de obicei ca root si este cel care deschide portul 25
- oprire: **postfix stop**
- recitirea configurarii: **postfix reload**
- fisiere de configurare:
  - **/etc/postfix/main.cf**. Poate fi citit/modificat direct sau prin intermediul utilitarului *postconf*
  - **/etc/postfix/master.cf** – fisierul de configurare al procesului coordonator *master*

12.2.3. Configurare

12.2.3.1. Fisierul main.cf

12.2.3.1.1. Sintaxa generala

Pentru configurarea serverului postfix este folosit fisierul */etc/postfix/main.cf*, ce contine un ansamblu de atribuiri *parametru = valoare*. Membrul drept al atribuirii poate fi:

- **o valoare simpla**, scrisa ca atare sau preluata de la un alt parametru de configurare:

```
myhostname    = mail.c4.ro
myorigin      = $mydomain          # folosirea valorii unui alt parametru, numit mydomain
```

- **o lista de valori**. Valorile pot fi separate prin virgula, spațiu sau TAB:

```
mydestination = $mydomain c1.ro c2.ro c3.ro
mydestination = $mydomain, c1.ro, c2.ro, c3.ro # cele doua linii sunt echivalente
```

- **o tabela de informație externa** – o modalitate de a memora liste lungi de inregistrari in locatii externe fisierului de configurare (subiectul este abordat în cadrul modulului de Linux Avansati)

```
local_recipient_maps = hash:/etc/postfix/destinatari
```

Toti parametrii de configurare au valori default, ceea ce reduce configurarea serverului la modificarea strictului necesar din totalul parametrilor disponibili.

12.2.3.1.2. Manipularea configurarii folosind postconf

Utilitarul **postconf** permite vizualizarea sau modificarea parametrilor de configurare ai serverului postfix. Desi configurarea de server se realizeaza in general prin editarea directa a fisierului *main.cf*, exista o intreaga serie de operatii pe care *postconf* le eficientizeaza:

postconf	afiseaza valorile curente ale tuturor parametrilor de configurare (lista poate fi apoi filtrata cu grep în scopul identificarii unui parametru cautat)
postconf *parametru*	afisarea valorii curente a parametrului solicitat
postconf –d	afisarea valorii default pentru toți parametrii postfix
postconf –d *parametru*	afisarea valorii default pentru parametrul solicitat
postconf –e *parametru='valoare'*	editarea valorii unui parametru – schimbarile sunt salvate in main.cf
postconf –n	afisarea parametrilor care au valoarea schimbată fata de default

Atentie! Dupa efectuarea de modificari in fisierul de configurare trebuie data comanda **postfix reload** (cu exceptia modificarii parametrului *inet_interfaces*, care impune restart de server).

12.2.3.2. Configurare de baza

Configurarea de pornire a serverului postfix include aspecte precum:

- numele DNS al statiei - se configureaza prin intermediul directivei **myhostname**. Acest nume este cel cu care se prezinta serverul in dialogul SMTP cu clientii sau cu alte servere si in mesajele de eroare generate (ex: bounce-uri). Valoarea directivei *myhostname* este de asemenea folosita ca default pentru diverse alte directive
- domeniul din care face parte statia - directiva folosita este **mydomain**; ea este necesara in cel putin doua situatii:
  - cand numele statiei este unul scurt (ex: directiva *myhostname* are valoarea *server*). In acest caz serverul formeaza numele DNS complet prin concatenarea valorii lui *myhostname* cu cea a lui *mydomain*
  - cand numele statiei ar crea ambiguitati (ex: pentru numele *mx1.a.b.ro* nu este clar daca domeniul este *a.b.ro* sau doar *b.ro*)
- sufixul adaugat adreselor incomplete - se stabileste cu directiva **myorigin**. Adresele incomplete se pot intalni in diferite locuri si situatii: in fisierul cu alias-uri, in mesaje - provenite fie din mesajele create de MUA-uri care nu respecta standardele (mai rar), fie de la utilitare de livrare a mailului din linia de comanda (vezi injectarea de mail local - 12.2.4) etc.

```
myhostname    = mx1.a.ro
mydomain      = a.ro
myorigin      = $mydomain
```

12.2.3.3. Criterii de acceptare a mesajelor

Atunci cand un client se conecteaza la serverul SMTP si incearca sa-i livreze un mesaj, serverul il va accepta doar daca configurarea sa o permite; acceptarea inseamna asumarea responsabilitatii pentru procesarea mesajului. Administratorul poate configura serverul sa autorizeze mesajele tinand cont de o multitudine de caracteristici ale clientului si/sau mesajului.

Reamintim scurt cazurile in care serverul accepta un mesaj:

- cand el gazduieste casutele domeniului destinatie; in acest caz, mesajele vor fi pasate unui MDA care va decide actiunea ce li se aplica
- cand serverul este configurat sa joace rolul de relay, caz in care el accepta mesajul insa il livreaza mai departe serverului de drept. Rolul de relay poate fi indeplinit:
  - pentru un set de domenii - decizia acceptarii se ia pe baza domeniului destinatie, administratorul configurand o lista de domenii pentru care serverul actioneaza ca relay
  - pentru un set de clienti - decizia acceptarii se ia pe baza caracteristicilor clientului, uzuale fiind adresa si/sau identitatea clientului (ex: username-ul furnizat la autentificare SMTP pe segmentul MUA-MSA)

Vom prezenta in continuare modul de configurare a serverului pentru fiecare dintre aceste cazuri.

12.2.3.4. Livrarea locala

12.2.3.4.1. Configurarea domeniilor pentru care serverul accepta mail si il livreaza local

Dupa cum s-a explicat, fiecare casuta postala corespunde unui user, care poate fi unul de sistem sau unul virtual. Serverul postfix foloseste doua module MDA diferite pentru cele doua tipuri de useri: executabilul numit *local* pentru useri de sistem, si cel numit *virtual* pentru useri virtuali; ele folosesc directive de configurare si baze de date separate.

Decizia de a folosi useri de sistem sau virtuali se ia per domeniu gazduit; exista liste separate de domenii care lucreaza cu useri de sistem sau cu useri virtuali, ele configurandu-se cu directive separate dupa cum urmeaza:

- **mydestination** - stabileste lista de domenii pentru care serverul accepta mail si incearca sa-l livreze in casute de useri de sistem
- **virtual_mailbox_domains** - stabileste lista de domenii pentru care serverul accepta mail si incearca sa-l livreze in casute de useri virtuali. Acest scenariu este unul mai complex si nu va fi prezentat in materialul de fata

Nota: *sintagma "livrare locala" nu trebuie luata mot-a-mot; faptul ca serverul accepta un mesaj si i-l paseaza MDA-ului nu inseamna ca acesta va fi livrat intr-o casuta de pe acelasi server (adresa poate fi un forward) si nici macar ca mesajul va fi livrat (in cazul in care adresa destinatie nu corespunde niciunei casute existente).*

Pentru a determina postfix sa accepte mesaje pentru mai multe domenii si sa incerce livrarea in casute de useri de sistem vom folosi o lista pe post de valoare a directivei *mydestination*:

```
mydestination = a.ro, b.ro, c.ro
```

Pentru un domeniu ce lucreaza cu useri de sistem, fiecare user din */etc/passwd* devine automat adresa e-mail. A crea o noua adresa in domeniu se reduce la a adauga un nou user de sistem, cu comenzile binecunoscute.

Nota: *in cazul in care dorim ca doar o parte dintre userii de sistem sa poata receptiona mail, putem folosi parametrul local_recipient_maps care specifica adresele valide ale domeniului. In acest fel putem exclude useri folositi pentru rularea de servere/servicii si care nu dorim sa aiba automat o adresa e-mail corespondenta.*

12.2.3.4.2. Locatia si formatul casutelor postale

Odata ce MDA a receptionat un mesaj de la modulul MTA si a identificat userul caruia ii corespunde casuta, trebuie sa livreze efectiv mesajul in casuta. Pentru aceasta este necesar sa cunoasca unde se afla casuta si care este formatul sau.

In privinta locatiei casutelor postale exista doua mari strategii posibile:

- plasarea tuturor casutelor intr-un singur director dedicat (de obicei */var/mail* sau */var/spool/mail*). Acest scenariu se implementeaza cu parametrul **mail_spool_directory** ce are ca valoare directorul dedesubtul caruia vor fi create casutele. Fiecare casuta va purta numele userului caruia ii apartine
- plasarea casutei fiecarui utilizator in directorul personal al acestuia (ex: *~/Maildir*). Exista în acest scop directiva **home_mailbox** a carei valoare este calea catre casuta (inclusiv numele casutei), relativa la directorul personal al userului; chiar dacă se foloseste o cale absolută, ea va fi creata tot dedesubtul directorului personal!

Formatul casutei sa stabileste dupa cum urmeaza:

- daca valoarea directivelor de mai sus se incheie cu /, livrarea va fi efectuata in format maildir
- daca valoarea nu se incheie cu /, formatul va fi implicit mbox

Iata cateva exemple, ce indica si locatia/formatul rezultante ale casutei userului *admin* (vom presupune ca directorul sau personal este in */home/admin*):

```
# toate casutele stau in /var/spool/mail, in format mailbox
mail_spool_directory = /var/spool/mail
# casuta userului admin va fi fisierul obisnuit /var/spool/mail/admin
```

```
# toate casutele stau in /var/spool/mail, in format maildir
mail_spool_directory = /var/spool/mail/                    # /-ul din final indica maildir
# casuta userului admin va fi directorul /var/spool/mail/admin, cu subdirectoarele new, cur si tmp
```

```
# livrare in format mailbox in directorul personal
home_mailbox = Mail
# casuta userului admin va fi fisierul obisnuit /home/admin/Mail
#  Acelasi efect l-am fi obtinut si daca scriam home_mailbox = /Mail, calea fiind oricum
#  considerata relativa la directorul personal; o greseala tipica este scrierea caii absolute,
#  de forma home_mailbox = /var/mail, casutele userilor fiind create ca urmare in ~/var/mail !
```

```
# livrare in format maildir in directorul personal
home_mailbox = Mailul_meu/                                 # /-ul din final indica maildir
# casuta userului admin va fi directorul /home/admin/Mailul_meu, cu subdirectoarele new, cur, tmp
```

Nota: *daca sunt definite ambele directive,* home_mailbox *este prioritar fata de* mail_spool_directory.

Fiecare casuta este creata automat de catre server la prima livrare in acea casuta. Procesul de livrare ruleaza cu UID-ul si GID-ul primar al userului destinatie. In consecinta trebuie configurate corect ownership-ul si permisiunile pe directorul in care se afla casuta, astfel incat crearea ei sa fie posibila:

- in cazul in care casutele sunt plasate in directoarele personale de useri, acestea din urma au deja ca owner userul in cauza si dispun de permisiuni suficiente
- in cazul plasarii tuturor casutelor intr-un director dedicat lor se pot folosi diferite strategii, insa ele presupun in general ca toti userii sa aiba drept de write pe acel director, astfel incat serverul sa le poata crea automat casutele. In acest fel, orice user ar putea sterge casutele celorlalti, si de aceea pe directorul in cauza va fi setata in general permisiunea speciala *sticky*

12.2.3.4.3. Lucrul cu alias-uri

Dupa cum s-a explicat, alias-urile sunt adrese virtuale ce se pot expanda intr-una sau mai multe actiuni posibile: livrare, forward, o combinatie de acestea sau diferite alte actiuni (salvare mesaj intr-un fisier, pasare catre un program extern etc).

Baza de date cu alias-uri este un fisier binar, care nu se editeaza direct. Administratorul de server defineste si modifica alias-urile intr-un fisier text, pe care mai apoi il compileaza folosind comanda **newaliases**, producand fisierul binar. Exista urmatoarele doua directive de configurare postfix in acest scop:

- **alias_maps** - stabileste locatia si formatul bazei de date cu alias-uri. Locatia sa implicita este de obicei */etc/aliases* sau */etc/postfix/aliases*. Alias-urile pot fi memorate in una sau mai multe surse de informatie (fisiere, baze de date externe etc)
- **alias_database** - stabileste lista de fisiere ce vor fi compilate ca urmare a comenzii *newaliases*. Aceasta directiva exista deoarece este posibil ca nu toate sursele de informatie listate in *alias_maps* sa fie fisiere. Valoarea default a directivei este de asemenea */etc/aliases*.

Atenţie! *După fiecare modificare adusă fisierului text, baza de date binara trebuie regenerata cu comanda* newaliases*!*

Fiecare linie din fisierul cu alias-uri este de forma urmatoare:

```
nume: destinatie1, destinatie2, ….
```

Numele reprezinta partea stanga a adresei virtuale destinatie (“username-ul”). Destinatiile posibile pot fi:

- **nume de useri locali**. Orice nume de user va primi ca sufix, la livrare, valoarea directivei *myorigin* pentru a se forma o adresa completa. Rezultatul este livrarea in casuta acelui user daca el exista
- **adrese complete**. Daca adresa este una locala (casuta se gaseste pe acelasi server) se va incerca livrarea la casuta; in cazul unei adrese externe rezulta o redirectionare catre un alt server
- **nume de alias-uri deja definite**. Sunt posibile alias-urile succesive multiple; exemplu: *user1* este alias catre *user2*, care la randul lui este alias catre *user3*

- **fisiere externe**, specificate sub forma caii catre ele. In acest fel se poate realiza fie livrarea directa intr-o casuta maildir sau mailbox (specificand sau nu /-ul de la finalul caii), sau se pot distruge mailurile trimitandu-le in */dev/null*
- **comenzi externe**, specificate sub forma **|comanda**. Util pentru autorespondere sau pentru analiza mesajelor (ex: antivirus, antispam)

Iata un exemplu de fisier cu alias-uri:

```
office :contact
contact:admin
admin:vasile
dan:ana, vlad@b.ro, elena@c.ro
ion:ion@a.ro,ion.pop@gmail.com
ina:/dev/null
sales:/var/spool/mail/vanzari/
```

Sa presupunem ca fisierul este definit pentru serverul ce gazduieste domeniile *a.ro* si *b.ro*, iar valoarea directivei *myorigin* este *a.ro*. In aceste conditii:

- mailul trimis catre *office@a.ro* circula pe filiera de alias-uri successive office → contact → admin → vasile si va fi livrat in casuta userului local *vasile* (care, nemaifiind alias, este expandat inainte de livrare in *vasile@a.ro*)
- mailul trimis catre *dan@a.ro* sau *dan@b.ro* (baza de date cu alias-uri nu face diferenta intre domenii) va fi expandat in 3 actiuni:
  - livrare catre userul local *ana* (care primeste ca sufix valoarea lui *myorigin* si devine *ana@a.ro*)
  - livrare catre userul local *vlad* (care are deja ca sufix unul dintre domeniile locale)
  - redirectionare catre adresa externa *elena@c.ro* (postfix va determina cu ajutorul DNS serverul responsabil pentru domeniul *c.ro* si ii va livra mailul)
- mailul trimis catre *ion@a.ro* sau *ion@b.ro* va fi atat livrat local in casuta userului *ion*, cat si forward-at catre *ion.pop@gmail.com*
- mailul trimis catre *ina@a.ro* sau *ina@b.ro* va fi distrus
- mailul trimis catre *sales@a.ro* sau *sales@b.ro* va fi livrat in casuta */var/spool/mail/vanzari*, in format maildir

Nota: *exista posibilitatea ca fiecare user local al unei statii sa-si defineasca propriile actiuni aplicate mesajelor primite, folosind fisierul ~/.forward, a carui fiecare linie are un format identic cu al partii drepte a unui alias din baza de date globala.*

12.2.3.5. Functionarea ca relay

12.2.3.5.1. Configurarea domeniilor pentru care serverul joaca rolul de relay

Atunci cand dorim ca serverul sa accepte mesaje pentru unul sau mai multe domenii, insa fara a incerca sa le livreze local (spre exemplu, in cazul unui backup MX) vom folosi directiva **relay_domains** pentru a specifica lista de domenii in cauza.

```
relay_domains = c.ro, d.ro
```

Serverul va accepta mesajele destinate acelor domenii si le va trimite mai departe catre serverele de drept.

12.2.3.5.2. Configurarea listei de adrese de clienti pentru care serverul joaca rolul de relay

Aceasta operatie este utila/necesara atunci cand dorim ca postfix sa joace rolul de MSA fara autentificare, permitand relay-ul doar in functie de adresele IP ale clientilor. Setul de clienti carora li se permite relay-ul trebuie sa fie unul bine controlat; in caz contrar se creeaza un *open relay*, situatie total contraindicata.

Directivele postfix cu care putem configura relay-ul pe baza adresei IP a clientului sunt:

- **mynetworks_style** – are ca valoare cateva categorii prestabilite de clienti:

- ***class*** – permite relay-ul tuturor clientilor aflati in aceeasi retea cu serverul, unde reteaua este determinata strict pe baza clasei de care apartine adresa serverului. Spre exemplu, daca serverul are adresa 192.168.1.10, aceasta este recunoscuta ca fiind o adresa de clasa C si in consecinta clientii carora li se va permite relay-ul vor fi cei din reteaua 192.168.1.0/24. Daca adresa ar fi fost 10.0.0.10, atunci clientii acceptati apartineau subnet-ului 10.0.0.0/8
- ***subnet*** – permite relay-ul numai clientilor din acelasi subnet cu serverul. Spre desosebire de cazul anterior, de aceasta data se tine cont de netmask-ul asociat adreselor IP ale serverului
- ***host*** – permite relay-ul numai daca clientul se conecteaza de pe aceeasi masina cu serverul

- **mynetworks** – folosit atunci cand se doreste deschiderea relay-ului pentru o lista de clienti care nu coincide cu niciunul dintre preset-urile de mai sus. Valoarea acestei directive este o lista de adrese/subnet-uri pentru care se va permite relay-ul:

```
mynetworks = 127.0.0.1, 10.0.0.0/16, 192.168.0.0.24
```

12.2.4. Verificarea corectei functionari si diagnosticarea problemelor

In scopuri de verificare si diagnosticare, administratorul de server trebuie sa aiba posibilitatea de a trimite mailuri prin intermediul serverului sau. Desi acest lucru poate fi realizat prin instalarea si configurarea unui client email grafic, este mai rapid si mai comod ca operatia sa se poata realiza din linia de comanda. Serverul postfix ofera un utilitar numit **sendmail** cu care se pot injecta mailuri in coada sa de mesaje (fara a mai trece prin modulul de server SMTP), urmand ca acestea sa fie apoi procesate conform algoritmului obisnuit. In forma cea mai simpla de utilizare, executabilul *sendmail* primeste in standard input mesajul ce trebuie transmis, ca in exemplul urmator:

```
student@server$ echo Mesaj de test | sendmail user1@a.ro
```

Nota: *in unele distributii executabilul sendmail este redenumit* ***sendmail.postfix*** *deoarece este instalat si executabilul* sendmail *ce apartine de serverul cu acelasi nume.*

Verificarea livrarii se poate realiza in cel putin doua moduri:

- in cazul unei livrari locale poate fi analizat direct continutul casutei. Spre exemplu, daca serverul a fost configurat cu *home_mailbox=Posta/*, atunci in urma mesajului trimis cu comanda de mai sus ar trebui sa gasim in directorul */home/user1/Posta/new* un fisier cu nume de forma *1449735872.V801I40ecdM140624.server* ce contine, pe langa un set de headere SMTP, textul transmis
- postfix consemneaza in loguri atat livrarile efectuate cu succes, cat si esecurile. Felul in care este distribuita informatia in loguri difera usor de la o distributie la alta; spre exemplu, in Fedora exista fisierul */var/log/maillog*, pe cand in Ubuntu exista doua fisiere - unul pentru erorile de configurare (*/var/log/mail.err*) si unul in care se inregistreaza rezultatul incercarilor de livrare (*/var/log/mail.log*)

12.3. Serverul IMAP/POP dovecot

12.3.1. Descriere generala

Dovecot este un server IMAP/POP programat de la bun inceput cu intentia de a fi cat mai sigur (site-ul sau, www.dovecot.org, enumera chiar cateva dintre tehnicile de programare folosite in acest scop). Serverul este modular - compus dintr-un ansamblu de utilitare care interactioneaza, fiecare dintre ele rulat cu privilegiile minime necesare rolului sau.

Dovecot este in foarte dese cazuri folosit in combinatie cu postfix, fiind capabil sa acceseze casutele create de acesta din urma daca este configurat corect.

12.3.2. Instalare

Instalarea poate fi efectuata, ca de obicei, din surse sau din pachete precompilate. Cel din urma caz solicita insa o mica atentionare: unele distributii divizeaza serverul in mai multe pachete. Spre exemplu, in Ubuntu exista un pachet numit *dovecot-core*, insa functionalitatea de server POP si IMAP este asigurata de pachetele aditionale *dovecot-pop3d* si *dovecot-imapd*; in plus exista pachete suplimentare pentru backend-urile de autentificare suportate de dovecot (*dovecot-mysql*, *dovecot-pqsql* etc)

12.3.3. Fisa serverului

Iata cateva elemente ce trebuie/merita cunoscute de la bun inceput despre serverul dovecot:

- fisier de configurare principal: **dovecot.conf**, dar vezi explicatiile de la 12.3.4.1
- executabil: **dovecot**, cu ajutorul caruia pot fi efectuate operatii precum:
  - rulare in foreground: **dovecot -F**. Aceasta optiune ruleaza serverul in foreground insa nu afiseaza automat si logurile pe ecran! Optiunea este insa utila pentru a putea opri serverul usor in etapa de configurare initiala
  - reincarcarea din mers a configurarii: **dovecot reload**
  - oprirea serverului: **dovecot stop**
- utilitar de management al configurarii: **doveconf**, ce face posibile operatii precum:
  - afisarea valorii tuturor parametrilor: **doveconf | less**
  - afisarea valorii unui anumit parametru: **doveconf *nume_parametru***
  - afisarea parametrilor care au valori schimbate fata de default: **doveconf -n**

Daca a pornit corect, serverul creeaza un proces principal (cel care deschide porturile 110 si 143) care ruleaza ca root, un proces de autentificare (care ruleaza ca root daca autentificarea are nevoie de drepturi de root – de exemplu daca baza de date cu parole este */etc/shadow*) si unul sau mai multe procese de login pentru fiecare din serverele pornite.

12.3.4. Configurare

12.3.4.1. Fisiere de configurare si formatul acestora

Configurarea se prezinta sub forma unuia sau mai multor fisiere, ce contin directive de doua feluri:

- directive simple:

```
parametru = valoare
```

- directive bloc:

```
nume_directiva {
      directiva_simpla1 = valoare1
      directiva_simpla2 = valoare2
}
```

Se intalnesc doua mari abordari ale structurarii configurarii:

- implicit configurarea serverului dovecot foloseste fisierul principal *dovecot.conf* alaturi de care se afla subdirectorul *conf.d/* ce contine fisiere de configurare de forma *10-mail.conf*, *20-imap.conf* etc. Fisierele din *conf.d* sunt incluse in fisierul principal folosind directiva *!include*. Numarul cu care incepe numele fiecarui fisier din *conf.d* determina ordinea in care sunt incluse fisierele secundare in cel principal. Acest mod de structurare permite gasirea usoara a locului in care trebuie editata o directiva (ex: daca vrem sa modificam aspecte legate de logging vom edita direct fisierul *10-logging* din *conf.d*)

```
student@server:~$ tree /etc/dovecot/
/etc/dovecot/
├── conf.d
│   ├── 10-auth.conf
│   ├── 10-director.conf
│   ├── 10-logging.conf
│   ├── 10-mail.conf
│   ├── 10-master.conf
│   ├── 10-ssl.conf
│   ├── 10-tcpwrapper.conf
│   ├── 15-lda.conf
│   ├── 15-mailboxes.conf
│   ├── 90-acl.conf
│   ├── 90-plugin.conf
│   ├── 90-quota.conf
│   ├── auth-checkpassword.conf.ext
│   ├── auth-deny.conf.ext
│   ├── auth-master.conf.ext
│   ├── auth-passwdfile.conf.ext
│   ├── auth-sql.conf.ext
│   ├── auth-static.conf.ext
│   ├── auth-system.conf.ext
│   └── auth-vpopmail.conf.ext
├── dovecot.conf
├── dovecot-dict-sql.conf.ext
├── dovecot.pem
├── dovecot-sql.conf.ext
├── private
└── README
```

- unele distributii opteaza pentru folosirea unui singur mare fisier de configurare *dovecot.conf*, ce include toate directivele necesare

Nota: *chiar si in cazul default, al structurarii configurarii in fisier principal si secundare, o directiva poate fi scrisa in orice fisier (cu exceptia directivelor care pot fi folosite numai in cadrul anumitor directive bloc).*

Toti parametrii au valori default, prezentate de obicei in fisiere sub forma de comentarii, si de aceea se vor decomenta numai liniile unde trebuie produse modificari.

12.3.4.2. Locatia si formatul casutelor postale

12.3.4.2.1. Discutie

Aceasta reprezinta cu siguranta una dintre cele mai importante setari dovecot, deoarece ea asigura corecta interactiune cu serverul SMTP. Este esential ca serverul dovecot sa cunoasca aceeasi locatie si format ca si postfix, in caz contrar nerealizandu-se “intalnirea” celor doua softuri in casutele postale.

Serverul IMAP permite crearea de casute secundare; dupa cum s-a explicat, acest lucru este necesar, avand in vedere ca mesajele raman stocate pe server si utilizatorul are nevoie sa si le organizeze convenabil. Avem asadar urmatoarele doua categorii de casute:

- **inbox-ul**. Este casuta principala, cea in care livreaza mesaje serverul SMTP (acesta neavand conceptul de casute secundare) si din care citeste serverul IMAP
- **casutele secundare**. Sunt casute separate de inbox, in care informatia este scrisa si citita doar de catre serverul IMAP, la cererea userului. Ele se vad in MUA-ul utilizatorului ca niste foldere

Dovecot reprezinta casutele secundare in sistemul de fisiere in mod diferit, in functie de formatul lor. Distingem urmatoarele situatii:

- atunci cand inbox-ul este in format maildir, casutele secundare sunt create sub forma de subdirectoare maildir chiar in interiorul maildir-ului de baza (alaturi de cele standard - *new*, *cur* si *tmp*)

- cand inbox-ul este in format mbox, casutele secundare sunt la randul lor fisiere mbox, insa locatia lor poate fi - si deseori trebuie - configurata de catre administrator; in anumite scenarii ele nu pot sta in acelasi director cu inbox-ul (detalii in sectiunile urmatoare)

In ambele scenarii de mai sus, directiva dovecot folosita pentru configurarea locatiei si formatului casutelor este **mail_location**. In cazul configurarii structurate ea se gaseste in **/etc/dovecot/conf.d/10-mail.conf**. Sintaxa sa generala este:

```
mail_location = format_casuta:cale[:parametri...]
```

Formatul de casuta poate fi *mbox* sau *maildir*. Felul in care se defineste calea si eventualii parametri va fi detaliat in sectiunile ce urmeaza.

12.3.4.2.2. Cazul casutelor in format maildir

Daca inbox-urile userilor sunt in format maildir, este suficienta specificarea locatiei lor si dovecot va sti automat unde sa plaseze casutele secundare. Avand in vedere ca valoarea directivei *mail_location* trebuie sa fie valabila pentru toti userii, in cadrul ei pot fi folosite urmatoarele secvente speciale:

- **~** - desemneaza calea catre directorul personal al userului
- **%u, %n** si **%d** - reprezinta adresa email a userului, username-ul (partea stanga a adresei), respectiv domeniul (partea dreapta a adresei). Spre exemplu, daca userul se autentifica prin IMAP folosind ca username adresa *user1@a.ro*, atunci %u va avea ca valoare *user1@a.ro*, %n va avea valoarea *user1* iar %d va avea ca valoare *a.ro*.

```
# Cazul in care casutele sunt plasate in directoarele personale ale userilor;
#   daca in postfix a fost configurat home_mailbox=Posta/ , atunci in dovecot vom scrie:
mail_location = maildir:~/Posta

# Cazul in care casutele sunt plasate intr-un director comun, fiecare casuta
#   avand numele userului de care apartine; daca in postfix a fost configurat
#   mail_spool_directory = /var/spool/mail/ , atunci in dovecot vom scrie:
mail_location = maildir:/var/spool/mail/%u
```

12.3.4.2.3. Cazul casutelor in format mbox

Atunci cand casutele sunt in format mbox, fiecare casuta secundara este un fisier mbox la randul sau. Avem doua cazuri:

- daca inbox-ul este stocat in directorul personal al userului, putem configura dovecot sa plaseze casutele secundare tot acolo, sub forma de mbox-uri aditionale. In general este recomandabil ca atat inbox-ul, cat si restul de casute sa fie stocate intr-un subdirector al directorului personal, pentru a nu se amesteca cu restul de fisiere aflate in homedir. Putem face acest lucru specificand ca valoare a directivei *mail_location* subdirectorul in care sa fie plasat intregul ansamblu de casute:

  ```
  mail_location = mbox:~/mail # subdirectorul in care sta casuta principala+cele secundare
  ```

  Dovecot numeste automat fisierul corespunzator casutei principale *inbox*. Aceasta inseamna ca in postfix ar trebui sa configuram *home_mailbox = mail/inbox*. Numele fisierului inbox poate fi schimbat cu parametrul INBOX (vezi mai jos)

- daca inbox-ul este stocat intr-un spatiu comun (ex: */var/mail*) atunci este necesara separarea casutelor secundare ale diferitilor useri; a le lasa pe toate in acelasi director ar duce, mai devreme sau mai tarziu, la conflicte de nume, cand un user va alege un nume de casuta secundara identic cu al altuia. Solutia este plasarea casutelor secundare ale fiecarui user intr-un director separat si specificarea lui in configurare; spre exemplu putem plasa casutele secundare in subdirectorul *~/mailuri*, inbox-urile ramanand in */var/spool/mail*:

  ```
  mail_location = mbox:~/mailuri:INBOX=/var/spool/mail/%n
  ```

Corespunzator acestei directive, in postfix vom configura numai *mail_spool_directory = /var/spool/mail* (postfix-ul nu are nicio tangenta cu casutele secundare, el livrand numai in inbox).

12.3.4.3. Autentificare

Un al doilea aspect major ce trebuie configurat la dovecot este acela al autentificarii. Spre deosebire de SMTP, unde autentificarea exista numai pe anumite segmente, protocoalele POP si IMAP functioneaza intotdeauna cu autentificare, deoarece ele ofera acces la informatiile confidentiale din casutele postale.

In dovecot configurarea se realizeaza cu ajutorul a doua directive bloc:

- **userdb** - stabileste unde este stocata baza de date cu conturi de utilizator
- **passdb** - stabileste baza de date cu parole

Ambele directive au ca principala setare interna directiva simpla **driver**. Iata modul de configurare ce foloseste bazele de date de sistem:

```
passdb{       # baza de date cu parole
  driver = shadow
}
userdb{       # baza de date cu useri
  driver = passwd
}
```

In cazul configurarii structurate, fisierele din *conf.d* sunt organizate astfel:

- pentru fiecare backend de autentificare suportat de dovecot exista un fisier in *conf.d* cu extensia ext (ex: *auth-system.conf.ext*, *auth-sql.conf.ext*) care poate fi editat pentru a face setarile necesare acelui backend
- fisierul *10-auth.conf* din *conf.d* contine la finalul sau o lista de directive *!include*, cate una pentru fiecare fisier dintre cele amintite la punctul anterior. Se vor comenta toate directivele *!include* cu exceptia celei corespunzatoare backend-ului dorit

12.3.4.4. Logging si debugging

Implicit serverul dovecot face logging prin intermediul daemonului centralizat din Linux, syslogd. In scopul diagnosticarii serverului, dovecot poate fi configurat sa mentina fisiere log separate, putandu-se solicita marirea gradului de detaliu al informatiilor consemnate:

```
# locatia fisierului log
log_path = /var/log/dovecot.log

# loguri detaliate de autentificare
auth_verbose = yes
auth_debug = yes
```

In cazul configurarii structurate, directivele de mai sus se gasesc in **/etc/dovecot/conf.d/10-logging.conf**.

Nota: *pentru o modalitate de a verifica rapid si usor corecta configurare a autentificarii si a locatiei/formatului casutelor consultati Anexa 1 a materialului.*

12.3.4.5. Relaxarea temporara a setarilor de securitate

Securitatea este deseori invers proportionala cu usurinta de utilizare si diagnosticare. In scopul unei usoare diagnosticari a serverului putem face urmatoarele compromisuri temporare:

- activarea autentificarii plaintext. Aceasta presupune transmiterea userului si parolei in clar de la client catre server; desi din punct de vedere al securitatii este o idee proasta (caci ele pot fi interceptate pe parcurs), aceasta setare permite diagnosticarea facila a serverului din linia de comanda (vezi 12.5)
- dezactivarea SSL. Desi o conexiune criptata intre clienti si server creste nivelul de securitate, ea impiedica diagnosticarea usoara din linia de comanda

```
# activare transmitere user/parola clear-text
disable_plaintext_auth = no
# in cazul configurarii structurate, directiva se gaseste in /etc/dovecot/conf.d/10-auth.conf

# dezactivare SSL
ssl = no
# in cazul configurarii structurate, directiva se gaseste in /etc/dovecot/conf.d/10-ssl.conf
```

12.4. BIBLIOGRAFIE

- Softuri componente ale sistemului e-mail:
  - M*A: https://en.wikipedia.org/wiki/Email_agent_(infrastructure)
  - MSA: https://tools.ietf.org/html/rfc6409
- Descrierea detaliata a sistemului de posta electronica: http://www.tcpipguide.com/free/t_TCPIPElectronicMailSystemConceptsandProtocolsRFC82.htm
- Documente IETF ce specifica protocoalele componente ale sistemului e-mail:
  - SMTP:
    - RFC 5321: https://tools.ietf.org/html/rfc5321
    - RFC 5322: https://tools.ietf.org/html/rfc5322
  - POP3: https://tools.ietf.org/html/rfc1939
  - IMAP: https://tools.ietf.org/html/rfc3501
- Documentatia oficiala postfix: http://www.postfix.org/documentation.html
- Documentatia oficiala dovecot: http://wiki2.dovecot.org/

12.5. ANEXA 1 - Diagnosticare rapida POP/IMAP din linia de comanda

Protocoalele POP si IMAP folosesc comenzi si raspunsuri clear-text, inteligibile pentru om si generabile de catre acesta, ceea ce deschide calea unei diagnosticari usoare a doua aspecte esentiale ale functionarii unui server POP/IMAP: autentificarea si locatia/formatul casutelor. Este suficient sa cunoastem cateva comenzi de baza ale protocoalelor in cauza.

Conectarea la serverul POP/IMAP se poate realiza din linia de comanda cu utilitarul **telnet**; acesta va transmite apoi ceea ce scrie userul direct catre aplicatia server.

Prezentam in continuare modul de conectare la server, autentificare si vizualizare mesaje pentru cele doua protocoale. Liniile in bold sunt comenzile scrise de utilizator, restul de linii reprezinta raspunsuri ale serverului.

12.5.1. POP3

```
student@server:~$ telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK Dovecot (Ubuntu) ready.
user user1
+OK
pass p1
+OK Logged in.
list
+OK 1 messages:
1 308
.
```

```
retr 1
+OK 308 octets
Return-Path: <root@a.ro>
X-Original-To: user1@a.ro
Delivered-To: user1@a.ro
Received: by mx.a.ro (Postfix, from userid 0)
      id B395D8047F; Fri, 11 Dec 2015 11:49:08 +0200 (EET)
Message-Id: <20151211094908.B395D8047F@mx.a.ro>
Date: Fri, 11 Dec 2015 11:49:08 +0200 (EET)
From: root@a.ro (root)

YYYYY
.
quit
+OK Logging out.
Connection closed by foreign host.
```

Daca dupa comenzile *user* si *pass* serverul raspunde cu ok inseamna ca autentificarea s-a realizat cu succes. Dupa cum se observa, userul si parola se transmit in clar; pentru ca serverul sa accepte acest lucru este necesar ca directiva dovecot *disable_plaintext_auth* sa aiba valoarea *no*.

Daca in urma comenzii *list* se vede corect lista de mesaje aflate in inbox, inseamna ca in dovecot au fost configurate corect locatia si formatul casutelor.

12.5.2. IMAP

```
student@server:~$ telnet localhost 143
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
* OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE STARTTLS AUTH=PLAIN]
Dovecot (Ubuntu) ready.
1 login user1 p1
1 OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE SORT SORT=DISPLAY
THREAD=REFERENCES THREAD=REFS THREAD=ORDEREDSUBJECT MULTIAPPEND URL-PARTIAL CATENATE UNSELECT
CHILDREN NAMESPACE UIDPLUS LIST-EXTENDED I18NLEVEL=1 CONDSTORE QRESYNC ESEARCH ESORT SEARCHRES
WITHIN CONTEXT=SEARCH LIST-STATUS SPECIAL-USE BINARY MOVE] Logged in
2 select inbox
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft \*)] Flags permitted.
* 3 EXISTS
* 0 RECENT
* OK [UNSEEN 2] First unseen.
* OK [UIDVALIDITY 1449818871] UIDs valid
* OK [UIDNEXT 5] Predicted next UID
* OK [NOMODSEQ] No permanent modsequences
2 OK [READ-WRITE] Select completed (0.000 secs).
3 fetch 1:* body[text]
* 1 FETCH (BODY[TEXT] {7}
YYYYY
)
* 2 FETCH (FLAGS (\Seen) BODY[TEXT] {4}
ZZ
)
* 3 FETCH (FLAGS (\Seen) BODY[TEXT] {4}
TT
)
3 OK Fetch completed.
4 logout
* BYE Logging out
4 OK Logout completed.
Connection closed by foreign host.
```

Fiecare comanda IMAP debuteaza cu un asa-numit tag (eticheta); atentie asadar ca numerele cu care incep comenzile de mai sus fac parte din comanda!

Prima comanda realizeaza login-ul, verificand astfel autentificarea.

Cea de-a doua comanda ne indica indirect daca locatia si formatul casutelor au fost configurate corect (3 EXISTS inseamna ca in casuta se gasesc 3 mesaje).

Comanda FETCH (care poate fi formulata intr-o multitudine de forme, in functie de informatia dorita) aduce corpul tuturor mesajelor de la 1 pana la ultimul.

Pentru comenzi suplimentare consultati http://donsutherland.org/crib/imap .