



# ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

## Τμήμα Πληροφορικής

### ΕΠΛ 421 - Προγραμματισμός Συστημάτων

#### ΑΣΚΗΣΗ 2 - Διαχείριση Μηνυμάτων SMTP και IMAP μέσω Προγραμματισμού Κελύφους Bash

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

Υπεύθυνος Εργαστηρίου: Παύλος Αντωνίου

Ημερομηνία Ανάθεσης: Παρασκευή 06/10/23

Ημερομηνία Παράδοσης: Τετάρτη 25/10/23 και ώρα 14:00 (3+ εβδομάδες)

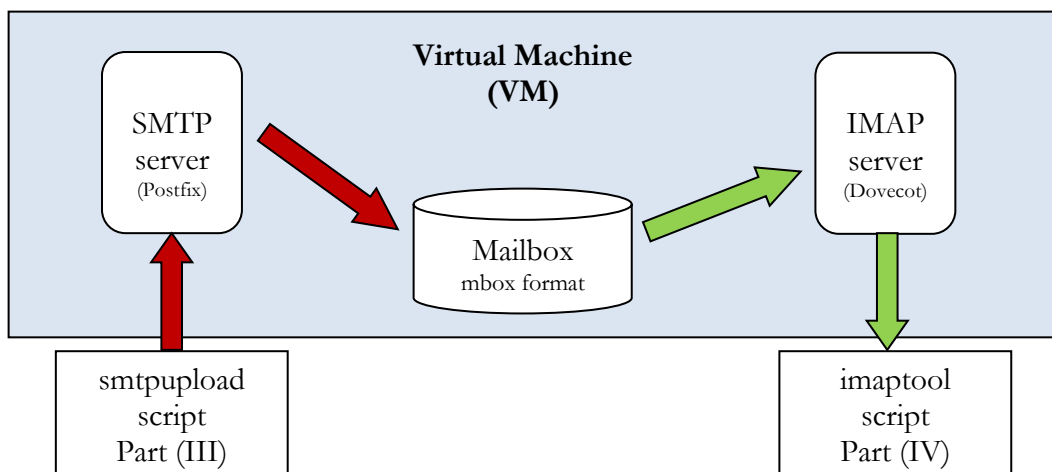
(η λύση να υποβληθεί σε zip μέσω του Moodle)

<https://www.cs.ucy.ac.cy/courses/EPL421>

#### I. Στόχος Άσκησης

Στόχος αυτής της άσκησης είναι η εξοικείωση με προχωρημένες τεχνικές προγραμματισμού στο κέλυφος Bash, και η εκτίμηση της ευκολίας με την οποία μπορεί κανείς να δημιουργήσει ένα σύνθετο σύστημα μέσω προγραμμάτων ωφελιμότητας (system utilities). Συγκεκριμένα, σε αυτή την άσκηση θα έχετε την ευκαιρία να χρησιμοποιήσετε έννοιες Διαχείρισης Συστημάτων (System Administration) και στη συνέχεια να κάνετε χρήση των εντολών του UNIX μέσω Προγραμματισμού Κελύφους (Bash Programming): εντολή *exec*, *πίνακες*, *συνθήκες ελέγχου*, *δομές επανάληψης*, *κανονικές εκφράσεις*, *επεξεργαστές ροών (sed, awk)* και *χρήση συναρτήσεων με τα προαναφερθέντα*.

Το θέμα της άσκησης είναι η υλοποίηση ενός προγράμματος αποστολής, ανάκτησης και ανάλυσης μηνυμάτων ηλεκτρονικού ταχυδρομείου (emails) πάνω από τα πρωτόκολλα SMTP (Simple Mail Transfer Protocol) και IMAP (Internet Message Access Protocol version 4) χωρίς την χρήση κάποιου έτοιμου εργαλείου (π.χ., Microsoft Outlook, Mozilla Thunderbird). Οι λειτουργίες του προγράμματος σας και το αναμενόμενο αποτέλεσμα περιγράφονται αναλυτικότερα στις παραγράφους III και IV. Το πιο κάτω διάγραμμα δίνει μια εποπτική θεώρηση των διαφόρων συστατικών και λειτουργιών του συστήματος.



## II. Προαπαιτήσεις

Τα πρωτόκολλα SMTP και IMAP ακολουθούν το μοντέλο πελάτη-εξυπηρετητή (client-server). Περισσότερες λεπτομέρειες θα δώσουμε πιο κάτω. Οι εξυπηρετητές ονομάζονται SMTP server και IMAP server. Για το σκοπό της άσκησης θα πρέπει να εγκαταστήσετε πάνω στη δική σας εικονική μηχανή ένα IMAP server και ένα SMTP server. Στα πλαίσια της άσκησης αυτής θα χρησιμοποιήσετε το πακέτο ανοικτού κώδικα Dovecot (<http://www.dovecot.org/>) το οποίο υλοποιεί IMAP και POP3 servers. Ο POP3 server δεν θα χρησιμοποιηθεί στα πλαίσια της άσκησης αυτής. Θα εγκαταστήσετε επίσης το πακέτο Postfix το οποίο υλοποιεί τον SMTP server.

### A) Εγκατάσταση IMAP Server (Dovecot)

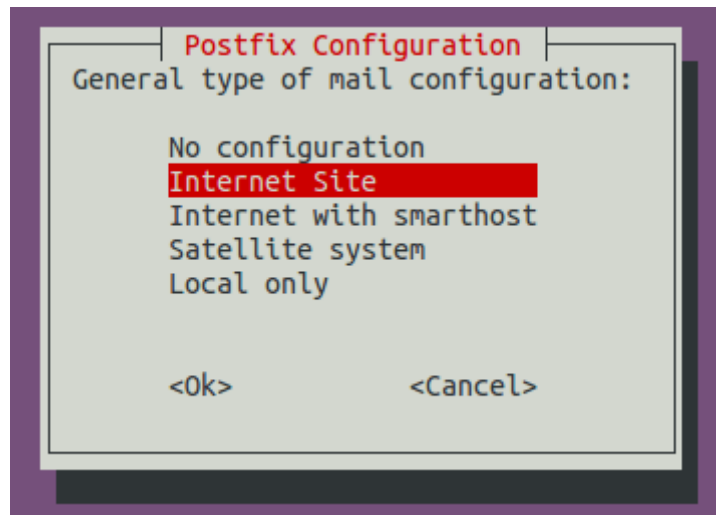
Οι πιο κάτω εντολές παρουσιάζουν πως μπορείτε να εγκαταστήσετε και να διαχειριστείτε τον Dovecot IMAP server:

- `sudo apt update`  
Εντολή για αναβάθμιση των repositories του ubuntu
- `sudo apt install dovecot-core dovecot-pop3d dovecot-imapd`  
Εντολή για εγκατάσταση του πακέτου dovecot.
- `service dovecot status`  
Εντολή που δείχνει την κατάσταση του dovecot. Θα λάβουμε την απάντηση ότι ο Dovecot τρέχει.
- `service dovecot stop`  
Η εντολή αυτή απενεργοποιεί το Dovecot.
- `vim /etc/dovecot/dovecot.conf`  
Η εντολή αυτή ανοίγει το βασικό configuration file του dovecot. Μέσα στον ίδιο κατάλογο /etc/dovecot θα βρούμε ένα άλλο κατάλογο conf.d/ που περιέχει περίπου 30 configuration files. Όλα μαζί συνενώνονται (μέσω της εντολής `include conf.d/*.conf` που βρίσκεται στο αρχείο /etc/dovecot/dovecot.conf) και αποτελούν το ολοκληρωμένο configuration για το Dovecot.  
Αν κάνουμε οποιαδήποτε αλλαγή σε κάποιο/α αρχείο/α αυτό/ά πρέπει να επανεκκινήσουμε το dovecot με την εντολή “`service dovecot restart`”.
- `vim /var/log/mail.log`  
Η εντολή αυτή ανοίγει το log file που περιέχει πληροφορίες σχετικά με τη λειτουργία του dovecot.

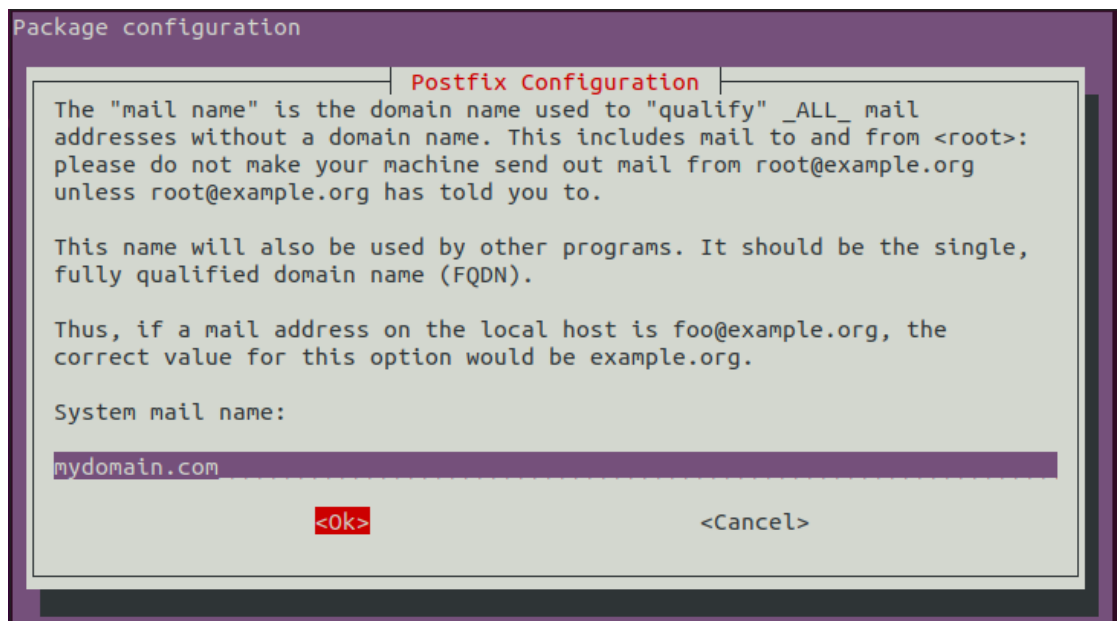
### B) Εγκατάσταση SMTP Server (Postfix)

Οι πιο κάτω εντολές παρουσιάζουν πως μπορείτε να εγκαταστήσετε και να διαχειριστείτε τον Dovecot IMAP server:

- `sudo apt install postfix -y`  
Εντολή για εγκατάσταση του πακέτου postfix. Στην πρώτη οθόνη που θα εμφανιστεί πατήστε tab (για να επιλεγθεί το <Ok >και μετά πατήστε Enter. Στην δεύτερη οθόνη αφήστε πατημένο το Internet site και πατήστε tab για να μεταφερθείτε στο <Ok> και στη συνέχεια πατήστε enter.



Η επιλογή Internet site επιτρέπει στον SMTP server να στέλνει και να λαμβάνει email από άλλους SMTP servers. Στην τρίτη οθόνη θα πρέπει να εισαγάγετε το domain name για τις διευθύνσεις email που θα υποστηρίζει ο SMTP server σας. Πιο συγκεκριμένα, αν το email σας είναι [pavlos@mydomain.com](mailto:pavlos@mydomain.com) τότε σε αυτή την τρίτη οθόνη θα πρέπει να εισαγάγετε το mydomain.com στη συνέχεια το tab και μετά enter για να προχωρήσετε.



- `service postfix status`  
Εντολή που δείχνει την κατάσταση του postfix. Θα λάβουμε την απάντηση ότι ο Postfix τρέχει.

### Γ) Έλεγχος θυρών υπηρεσιών

- `nmap localhost`  
Μετά την εκκίνηση του dovecot και του postfix μπορείτε να εκτελέσετε την εντολή `nmap localhost` για να δείτε ποιες υπηρεσίες τρέχουν στο VM σας αλλά και σε ποια θύρα (port) ακούνε. Αν το εργαλείο nmap δεν είναι εγκατεστημένο, να το εγκαταστήσετε μέσω του εργαλείου `apt install`. Μετά την εντολή `nmap localhost` παίρνουμε τα πιο κάτω:

PORT	STATE	SERVICE	
22/tcp	open	ssh	
<b>25/tcp</b>	<b>open</b>	<b>smtp</b>	Postfix package
80/tcp	open	http	
110/tcp	open	pop3	} Dovecot package
<b>143/tcp</b>	<b>open</b>	<b>imap</b>	
993/tcp	open	imaps	
995/tcp	open	pop3s	

Σημείωση: Πιθανόν να τρέχουν κι άλλες υπηρεσίες σε εσάς π.χ. ipp (στη θύρα 631/tcp) ή/και mysql (στη θύρα 3306/tcp).

### Δ) Ρυθμίσεις Mailbox για IMAP (mbox format)

Στο αρχείο `/etc/dovecot/conf.d/10-mail.conf` μπορούμε να ορίσουμε σε ποιο κατάλογο θα αποθηκεύονται τα μηνύματα ηλεκτρονικού ταχυδρομείου των χρηστών. Στο αρχείο αυτό ορίζουμε τη παράμετρο `mail_location` να είναι:

```
mail_location = mbox:~/mail:INBOX=/var/mail/%u
```

Αν αυτό υπάρχει στο αρχείο σας, τότε δεν το τροποποιείτε.

Σύμφωνα με τη μορφή αποθήκευσης μηνυμάτων [mbox](#) όλα τα αρχεία ηλεκτρονικού ταχυδρομείου για κάθε χρήστη ξεχωριστά θα αποθηκεύονται σαν **ένα αρχείο απλού κειμένου** (plaintext) στον κατάλογο `/var/mail`. Το όνομα του αρχείου θα είναι το όνομα του χρήστη (%u). Το αρχείο θα δημιουργηθεί αυτομάτως όταν ο χρήστης λάβει το πρώτο του email.

Εκτός από τη μορφή `mbox` υπάρχει και η μορφή [maildir](#) κατά την οποία κάθε αρχείο ηλεκτρονικού ταχυδρομείου αποθηκεύεται σαν ξεχωριστό αρχείο πάνω στο δίσκο με προκαθορισμένη [μορφή](#). Παρόλο που το `maildir` είναι νεότερο από το `mbox`, στην άσκηση αυτή θα χρησιμοποιήσουμε το πρώτο για απλούστευση (για να είναι συμβατό το format με τη προεπιλεγμένη μορφή αποθήκευσης μηνυμάτων του SMTP server.)

Σημείωση: Η μορφοποίηση των αρχείων Email στο διαθέτη του Τμήματος Πληροφορικής είναι σε `maildir`.

### Ε) Πιστοποίηση Ταυτότητας Χρηστών (Authentication) στο IMAP

Το αρχείο `/etc/dovecot/conf.d/10-auth.conf` περιέχει πληροφορίες σχετικά με την πιστοποίηση ταυτότητας (authentication) των χρηστών που θα χρησιμοποιούν την υπηρεσία που προσφέρει το Dovecot. Ως χρήστες της υπηρεσίας, μπορούν να χρησιμοποιηθούν οι χρήστες του συστήματος UNIX (default option) ή χρήστες που βρίσκονται καταχωρημένοι σε βάση δεδομένων SQL κτλ. Στην άσκηση αυτή, θα χρησιμοποιήσουμε τους χρήστες του συστήματος.

**Προσθήκη Χρήστη στο Unix:** Για το σκοπό αυτό θα δημιουργήσουμε ένα νέο χρήστη, έστω `imapuser`. Με την εντολή: `sudo adduser imapuser` δημιουργείται ο χρήστης του συστήματος UNIX και αυτομάτως δημιουργείται και ο προσωπικός του κατάλογος `imapuser/` μέσα στον κατάλογο `home`. Ο χρήστης αυτός είναι ιδιοκτήτης του καταλόγου αυτού (έχει όλα τα δικαιώματα). Την ώρα της δημιουργίας του χρήστη πρέπει να δοθεί ο κωδικός. Για ευκολία αν θέλετε βάλτε τον ίδιο κωδικό με το VM. Δεν είναι ανάγκη να συμπληρωθούν οι ζητούμενες πληροφορίες (Full Name, κτλ) οπότε πατάτε `enter` και προχωράτε μέχρι την τελευταία ερώτηση «Is the information correct? στην οποία πρέπει να απαντήσετε `y`.

Επίσης στο αρχείο αυτό μπορούμε να ορίσουμε αν θέλουμε να επιτρέπεται ή όχι η πιστοποίηση ταυτότητας με χρήση απλού κειμένου (δείτε μεταβλητή `disable_plaintext_auth`). Για λόγους απλοποίησης, στα πλαίσια της άσκησης αυτής θέλουμε να επιτρέπεται μη κρυπτογραφημένη σύνδεση (δείτε επόμενη ενότητα III) στην οποία (ως προεπιλογή) δεν επιτρέπεται πιστοποίηση ταυτότητας απλού κειμένου. **Για να αποφύγουμε αυτή την προεπιλογή θα πρέπει να θέσουμε στο υπό εξέταση αρχείο τη μεταβλητή `disable_plaintext_auth = no`.**

```
csdeptucy@ubuntu: ~
# Disable LOGIN command and all other plaintext authentications unless
# SSL/TLS is used (LOGINDISABLED capability). Note that if the remote IP
# matches the local IP (ie. you're connecting from the same computer), the
# connection is considered secure and plaintext authentication is allowed.
# See also ssl=required setting.
disable_plaintext_auth = no
```

**Σημείωση:** Στο αρχείο `/etc/dovecot/conf.d/10-master.conf` μπορούμε να δούμε πληροφορίες όπως σε ποιες θύρες ακούνε οι διεργασίες IMAP & POP3 servers αλλά και άλλες διεργασίες που μπορούν να συνεργαστούν με το Dovecot. Στην παρούσα φάση δεν θα τροποποιήσουμε κάτι στο αρχείο αυτό.

Επανεκκινούμε τον Postfix server: `service postfix restart`

### III. Αποστολή Ηλεκτρονικών Μηνυμάτων μέσω πρωτοκόλλου SMTP

Το πρωτόκολλο SMTP επιτρέπει την αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου από είναι σύστημα πελάτη SMTP (π.χ. Mozilla Thunderbird) σε ένα εξυπηρετητή SMTP ο οποίος παραδοσιακά «ακούει» στο TCP port 25. Σκοπός της ενότητας αυτής είναι η υλοποίηση μιας εντολής (`smtpupload`) η οποία όταν εκτελείται να «ανεβάζει» σε ένα λογαριασμό ηλεκτρονικού ταχυδρομείου (email account) ένα αριθμό μηνυμάτων ηλεκτρονικού ταχυδρομείου που θα είναι αποθηκευμένα τοπικά σε ένα κατάλογο. Μπορείτε να κατεβάσετε ένα αρχείο zip που περιέχει τον κατάλογο με τα email (`as2-supplementary.zip`).

#### A) Παράδειγμα Μηνύματος Email (Header + Body)

Ένα μήνυμα ηλεκτρονικού ταχυδρομείου έχει την ακόλουθη μορφή (λάβετε υπόψη ότι μπορεί να περιέχονται και άλλα πεδία μέσα στην επικεφαλίδα):

Content-Type: text/plain; charset=ISO-8859-1 Content-Disposition: inline Content-Transfer-Encoding: quoted-printable MIME-Version: 1.0 Subject: Welcome to EPL371 Date: Tue, 02 Feb 2021 15:31:58 GMT	<b>Header</b>
Welcome to EPL421!  In this course, students will learn to develop complex system-level software in the C programming language while gaining an intimate understanding of the UNIX operating system (and all OS that belong to this family, such as Linux, the BSDs, and even Mac OS X) and its programming environment.	<b>Body</b>

Ας δούμε τι γίνεται όταν κάποιος θελήσει να συνδεθεί με τον προσωπικό του λογαριασμό ηλεκτρονικού ταχυδρομείου και να αποστείλει ένα μήνυμα, χωρίς την χρήση γραφικού φυλλομετρητή (webmail) ή άλλων εξιδεικευμένων εργαλείων π.χ. Microsoft Outlook, Mozilla Thunderbird κτλ.

## B) Επικοινωνία με SMTP μέσω File Descriptors

Θέλουμε να στείλουμε ένα email που να πάει στο ταχυδρομικό κουτί του imapuser (/var/mail/imapuser).

```
# Άνοιξε ένα τερματικό και εκτέλεσε την ακόλουθη εντολή, η οποία ανοίγει ένα tcp socket
(ένα κανάλι επικοινωνίας) με τον εξυπηρετητή μηνυμάτων ηλεκτρονικού ταχυδρομείου πάνω
στο VM (VM IP address ή localhost αν το script θα είναι πάνω στο VM σας), στην θύρα 25,
για ανάγνωση/γγραφή. Με την εντολή αυτή δημιουργείται ο χειριστής του socket #5 (socket
descriptor) μέσω του οποίου θα γίνεται η αποστολή και λήψη δεδομένων προς και από το
socket.
exec 5<>/dev/tcp/localhost/25

# Αρχικοποίησε την επικοινωνία με τον SMTP server.
echo "HELO your_username" >&5

# Στη συνέχεια μπορούμε να αποστέλλουμε εντολές του πρωτοκόλλου SMTP (βλέπε πιο
κάτω) για να έχουμε πρόσβαση σε μηνύματα ηλεκτρονικού ταχυδρομείου, ακολουθώντας τα
παράκάτω βήματα:
# 1) Προσδιορίστε τον αποστολέα του email. Μπορείτε να βάλετε οποιοδήποτε email
address αλλά και το δικό σας mail address π.χ. imapuser@mydomain.com).
# 2) Προσδιορίστε τον παραλήπτη του email. Πρέπει να είναι το imapuser@mydomain.com
για να πάει στον inbox του imapuser. Αν βάλετε παραλήπτη κάποιον που έχει το ίδιο domain
με εσάς π.χ. mydomain.com και δεν ανήκει στους χρήστες της μηχανής π.χ.
pavlos@mydomain.com τότε θα απορριφθεί από τον τοπικό SMTP server. Αν βάλετε
παραλήπτη κάποιο χρήστη άλλου λογαριασμού π.χ. antoniou.pavlos@ucy.ac.cy τότε ο
SMTP server θα προσπαθήσει να στείλει το μήνυμα στον SMTP server που εξυπηρετεί το
domain name ucy.ac.cy (δεν θέλουμε αυτό).
# 3) Αρχικοποίηση αποστολής περιεχομένου μηνύματος (data)
# 4) Αποκοπή επικεφαλίδας (header) ηλεκτρονικού μηνύματος (από αυτά που βρίσκονται
μέσα στον κατάλογο) και αποστολή της στον εξυπηρετητή
# 5) Αποκοπή σώματος (body) ηλεκτρονικού μηνύματος και αποστολή του
# 6) Αποστολή τέλους μηνύματος (για τερματιστεί η αποστολή κάθε μηνύματος πρέπει να
σταλεί η τελεία .)

# Μετά από κάθε εντολή μπορούμε να εκτυπώνουμε το την απάντηση του server στην οθόνη
read line <&5
echo $line
```

Στο τέλος του προγράμματος θα πρέπει να κλείσετε το input/output redirection του socket, για να απελευθερώσετε τον File Handler #5.

```
# Κλείσε το output redirection για το socket
exec 5>&-
# Κλείσε το input redirection για το socket
exec 5<&-
```

## Γ) SMTP Εντολές

Κάποιες βασικές εντολές που χρησιμοποιούνται σε μια σύνδεση SMTP είναι οι ακόλουθες:

HELO <username>	Ο χρήστης αρχικοποιεί την επικοινωνία και προσδιορίζει τον εαυτό του, συνήθως δηλώνοντας το domain name του
MAIL FROM: <email-addr>	Προσδιορισμός αποστολέα (οτιδήποτε email address)
RCPT TO: <email-addr>	Προσδιορισμός παραλήπτη (πρέπει να είναι <code>imapuser@mydomain.com</code> )
DATA	Μετά την εντολή αυτή, ο πελάτης θα αποστείλει το περιεχόμενο του ηλεκτρονικού μηνύματος
QUIT	Τερματισμός σύνδεσης

Αξίζει να σημειωθεί ότι κατά την αποστολή μηνυμάτων στον SMTP server (που είναι εγκατεστημένος στο VM σας) θα πρέπει η διεύθυνση ηλεκτρονικού ταχυδρομείου του παραλήπτη που δίδεται μετά την εντολή RCPT TO: να είναι η **imapuser@mydomain.com**. Ο server παραλαμβάνοντας ένα μήνυμα εξετάζει το domain name της ηλεκτρονικής διεύθυνσης (αυτό που είναι μετά το @) για να εξακριβώσει το domain που ανήκει ο χρήστης. Αν το domain name είναι σε αυτά που εξυπηρετεί ο server, ο server θα αποθηκεύσει τοπικά το μήνυμα (στο mailbox του χρήστη μέσα στο VM). Αν η διεύθυνση ηλεκτρονικού ταχυδρομείου του παραλήπτη είναι της μορφής [user@example.com](mailto:user@example.com) τότε ο SMTP server θα προωθήσει το μήνυμα στον υπεύθυνο SMTP server που εξυπηρετεί το domain name example.com.

## Δ) Ζητούμενο Μέρους Άσκησης:

Η νέα εντολή θα εκτελείται όπως φαίνεται παρακάτω:

```
./smtpupload emaildir smtpserver username
```

όπου emaildir είναι ο κατάλογος που θα βρίσκονται αποθηκευμένα τα μηνύματα (αυτά που θα βρείτε στο as2-supplementary.zip), smtpserver είναι ο εξυπηρετητής ηλεκτρονικού ταχυδρομείου (π.χ. localhost) και username το όνομα χρήστη για το email account (imapuser).

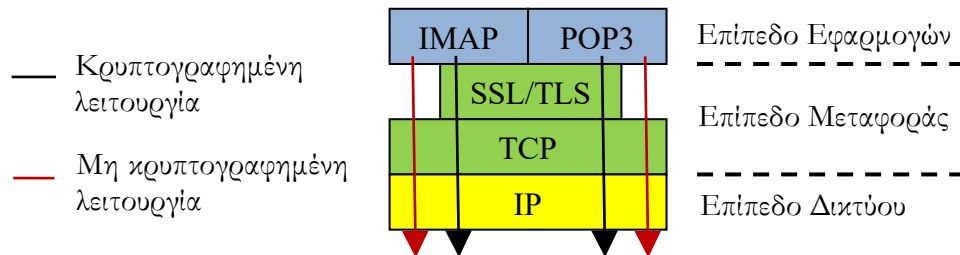
## IV. Ανάκτηση Ηλεκτρονικών Μηνυμάτων μέσω πρωτοκόλλου IMAP

Σε αυτή την ενότητα της εκφώνησης θα δούμε πως μπορεί κανείς να ανακτήσει μηνύματα ηλεκτρονικού ταχυδρομείου μέσω του κελύφους Bash υλοποιώντας την εντολή **imaptool**. Αρχικά θα δείξουμε πως επιτυγχάνεται σύνδεση μεταξύ πελάτη και εξυπηρετητή σε υψηλό επίπεδο, στη συνέχεια θα παρουσιάσουμε τις σημαντικότερες εντολές του πρωτοκόλλου **IMAP 4rev1** και τέλος θα δείξουμε πώς να εγκαθιδρύσετε ένα IMAP κανάλι επικοινωνίας μέσω εντολών του κελύφους Bash. Στόχος μας ΔΕΝ είναι να επεξηγήσουμε διεξοδικά το πρωτόκολλο μεταφοράς αρχείων IMAP (Request For Comments 3501: <https://tools.ietf.org/html/rfc3501>), διότι αυτό είναι το αντικείμενο μελέτης άλλων μαθημάτων.



## A) Κρυπτογραφημένες έναντι Μη-Κρυπτογραφημένες Συνδέσεις

Η πρόσβαση σε έναν εξυπηρετητή IMAP μπορεί να είναι είτε μη κρυπτογραφημένη ή κρυπτογραφημένη. Στη μη-κρυπτογραφημένη λειτουργία, τα δεδομένα που ανταλλάσσονται μεταξύ πελάτη και εξυπηρετητή είναι σε μορφή απλού κειμένου (plaintext) και υπόκεινται σε υποκλοπή από κακόβουλους χρήστες. Στην κρυπτογραφημένη λειτουργία, τα δεδομένα που ανταλλάσσονται εικάζονται να είναι κρυπτογραφημένα και δύσκολα μπορούν να αποκρυπτογραφηθούν από κακόβουλους χρήστες. Το Dovecot υποστηρίζει και τις δύο λειτουργίες. Για σκοπούς κρυπτογραφημένης λειτουργίας το Dovecot υποστηρίζει λειτουργία του πρωτοκόλλου IMAP (και POP3) πάνω από τα πρωτόκολλα SSL (Secure Socket Layer) / TLS (Transport Layer Security) όπως φαίνεται πιο κάτω:



Όπως μπορείτε να δείτε στο αρχείο **10-master.conf**, η μη-κρυπτογραφημένη λειτουργία υποστηρίζεται από τη διεργασία που ακούει στη θύρα 143 ενώ η κρυπτογραφημένη υπηρεσία από τη διεργασία που ακούει στην θύρα 993.

Το UNIX terminal δίνει δυνατότητα:

(α) **μη-κρυπτογραφημένης επικοινωνίας** με τον IMAP server μέσω του εργαλείου telnet στο terminal όπως με την πιο κάτω εντολή:

```
telnet localhost 143
```

(β) **κρυπτογραφημένης επικοινωνίας** μέσω του εργαλείου OpenSSL που υποστηρίζει τα πρωτόκολλα SSL/TLS στο terminal όπως με την πιο κάτω εντολή:

```
openssl s_client -connect localhost:993
```

Στην άσκηση αυτή, θα χρησιμοποιήσουμε την **μη-κρυπτογραφημένη λειτουργία** και θα υλοποιήσουμε το δικό μας πρόγραμμα-πελάτη (**χωρίς** τη χρήση των πιο πάνω εργαλείων).

## B) Τρόπος Λειτουργίας IMAP Πρωτοκόλλου

Το πρωτόκολλο IMAP λειτουργεί με βάση το μοντέλο πελάτη-εξυπηρετητή (client-server) επιτρέποντας στο πρόγραμμα-πελάτη (client) την πρόσβαση και διαχείριση αρχείων ηλεκτρονικού ταχυδρομείου που βρίσκονται στον εξυπηρετητή (server). Το πρωτόκολλο IMAP επιτρέπει τη διαχείριση κουτιών ηλεκτρονικού ταχυδρομείου (mailboxes) που στην ουσία είναι απομακρυσμένοι κατάλογοι μηνυμάτων (remote message folders) με τρόπο που είναι λειτουργικά ισοδύναμος με διαχείριση τοπικών καταλόγων (local folders).

Το πρωτόκολλο IMAP 4rev1 υποθέτει αξιόπιστη μεταφορά δεδομένων όπως αυτήν που παρέχεται από το πρωτόκολλο TCP. Αρχικά, ο εξυπηρετητής IMAP ανοίγει τις θύρες 143 και 993 περιμένοντας έναν πελάτη IMAP να συνδεθεί. Στη συνέχεια ο πελάτης ξεκινά μια νέα σύνδεση από μια τυχαία θύρα προς την θύρα 143 του εξυπηρετητή (για μη κρυπτογραφημένη



σύνδεση). Μόλις γίνει η σύνδεση, ο εξυπηρετητής στέλνει ένα χαιρετισμό, και ακολουθούν αλληλεπιδράσεις μεταξύ πελάτη και εξυπηρετητή. Κάθε αλληλεπίδραση αποτελείται από (α) μια εντολή πελάτη, (β) τα δεδομένα που επιστρέφει ο εξυπηρετητή και (γ) ένα μήνυμα ολοκλήρωσης αποστολής δεδομένων από τον εξυπηρετητή. Όλες οι αλληλεπιδράσεις ποτ αποστέλλονται από τον πελάτη και τον εξυπηρετητή είναι στη μορφή γραμμών, δηλ. συμβολοακολουθίες (string) που τελειώνουν με CRLF: \r (carriage return, CR) και \n (line feed, LF).

Η εντολή του πελάτη αρχικοποιεί μια λειτουργία. Κάθε εντολή πελάτη ξεκινά με ένα αναγνωριστικό (συνήθως μια μικρή συμβολοακολουθία, π.χ. A0001) το οποίο καλείται ετικέτα (tag). Για κάθε εντολή ο πελάτης δημιουργεί διαφορετική ετικέτα. Ο πελάτης πρέπει να ακολουθεί αυστηρά τη σύνταξη των εντολών που προδιαγράφονται στο RFC3501. Θεωρείται συντακτικό λάθος η αποστολή εντολών με ελλιπή ή επιπλέον κενά ή ορίσματα.

Ο εξυπηρετητής διαβάζει την εντολή του πελάτη, ελέγχει την εντολή και τα ορίσματά της και αποστέλλει πίσω δεδομένα και απόκριση ολοκλήρωσης αποτελεσμάτων. Η απόκριση ολοκλήρωσης αποτελεσμάτων υποδεικνύει την επιτυχημένη ή αποτυχημένη έκβαση της εντολής του πελάτη και ξεκινά με την ίδια ετικέτα με την οποία ξεκινούσε η εντολή του πελάτη. Αν υπάρχουν περισσότερες από μια εντολές πελάτη σε εξέλιξη, η ετικέτα στην απόκριση του εξυπηρετητή προσδιορίζει την εντολή πελάτη στην οποία αντιστοιχεί η απόκριση. **Υπάρχουν 3 πιθανές αποκρίσεις ολοκλήρωσης που στέλνονται από τον εξυπηρετητή: OK (που δηλώνει επιτυχία), NO (που δηλώνει αποτυχία) και BAD (που δηλώνει λάθος όπως μη αναγνωρισθείσα εντολή ή λανθασμένη σύνταξη εντολής). Το πρόγραμμα πελάτη που θα υλοποιήσετε θα πρέπει να είναι σε θέση να δεχθεί οποιαδήποτε από τις 3 αποκρίσεις του εξυπηρετητή.**

## Γ) Εντολές IMAP

Οι εντολές που περιγράφονται πιο κάτω είναι ένα υποσύνολο των εντολών που υποστηρίζει το πρωτόκολλο IMAP 4rev1, με τις οποίες θα ασχοληθούμε στην παρούσα άσκηση.

### LOGIN [user name] [password]

Εντολή αρχικοποίησης της σύνδεσης με καθορισμό ονόματος χρήστη και κωδικού.

**Παράδειγμα:** C: a001 LOGIN SMITH SESAME  
S: a001 OK [CAPABILITY ... ] Logged in

### LIST [reference name] [mailbox name]

Εντολή που παρουσιάζει τους καταλόγους του mailbox (μοιάζει με τη λειτουργία της εντολής κελύφους ls), χωρίς να δείχνει τα περιεχόμενα τους (τα μηνύματα που περιέχουν).

**Παράδειγμα:** C: A103 LIST "" "\*"  
S: \* LIST (\HasChildren \UnMarked) "/" "INBOX"  
S: \* LIST (\NoInferiors \UnMarked) "/" "INBOX/epl421"  
S: A103 OK List completed.

### SELECT [mailbox name]

Εντολή που επιτρέπει την επιλογή ενός mailbox για να έχουμε πρόσβαση στα μηνύματα που περιέχει (μοιάζει με τη λειτουργία της εντολής κελύφους cd INBOX).

**Παράδειγμα:** C: A142 SELECT INBOX

```

S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S: * OK [PERMANENTFLAGS (\Answered \Flagged \Deleted
\Seen \Draft \*)] Flags permitted.
S: * 30 EXISTS
S: * 4 RECENT
S: * OK [UIDVALIDITY 1455203974] UIDs valid
S: * OK [UIDNEXT 31] Predicted next UID
S: * OK [HIGHESTMODSEQ 1] Highest
S: A142 OK [READ-WRITE] Select completed.

```

## CLOSE

Εντολή που επιστρέφει την κατάσταση της σύνδεσης από selected σε authenticated. Δεν λαμβάνει ορίσματα.

**Παράδειγμα:**      **C: A341 CLOSE**  
                          **S: A341 OK CLOSE completed**

## FETCH [sequence set] [message data item names or macro]

Εντολή που ανακτά δεδομένα που σχετίζονται με κάποιο μήνυμα που είναι μέσα στο επιλεγθέν mailbox. Το sequence set (πρώτο όρισμα) ορίζει τα μηνύματα ή την ακολουθία μηνυμάτων που θα ανακτηθούν. Για παράδειγμα, το sequence set 2:4 αναφέρεται στα μηνύματα 2 έως και 4, με την σειρά που βρίσκονται μέσα στο mailbox. Το δεύτερο όρισμα καθορίζει το μέρος του μηνύματος που θα ανακτηθεί. Για παράδειγμα, η αναφορά σε «(BODY[HEADER.FIELDS (SUBJECT)])» φέρνει μόνο το subject μέρος της κεφαλίδας των μηνυμάτων 2:4.

**Παράδειγμα:**      **C: A654 FETCH 2:4 (FLAGS BODY[HEADER.FIELDS (DATE FROM)])**  
                          **S: \* 2 FETCH ....**  
                          **S: \* 3 FETCH ....**  
                          **S: \* 4 FETCH ....**  
                          **S: A654 OK FETCH completed**

## LOGOUT

Εντολή αποσύνδεσης. Δεν λαμβάνει κανένα όρισμα και μπορεί να εκτελεστεί από οποιαδήποτε κατάσταση σύνδεσης. Ο εξυπηρετητής ΠΡΕΠΕΙ να στείλει την απόκριση BYE χωρίς ετικέτα πριν την απόκριση OK (με ετικέτα) και να κλείσει τη σύνδεση.

**Παράδειγμα:**      **C: A023 LOGOUT**  
                          **S: \* BYE IMAP4rev1 Server logging out**  
                          **S: A023 OK LOGOUT completed**  
                          (Εξυπηρετητής και πελάτης κλείνουν τη σύνδεση)

## Δ) Ζητούμενο Μέρους Άσκησης:

Σε αυτό το μέρος της άσκησης καλείστε να υλοποιήσετε τις IMAP λειτουργίες πάνω στα αρχεία που βρίσκονται στον IMAP server και στη συνέχεια να αναλύσετε τα ανακτώμενα αρχεία. Η νέα εντολή θα εκτελείται όπως φαίνεται παρακάτω:

Πρότυπο εντολής: **./imaptool [options] imapserver username**

Το πρόγραμμά σας θα συνδέεται στον IMAP server και θα αποκτά πρόσβαση στο λογαριασμό του χρήστη με το δοθέν username. Η εντολή θα πρέπει να απαιτεί από τον χρήστη να δώσει το password του. Στην συνέχεια, ανάλογα με τα options θα γίνεται ανάκτηση-ανάλυση των αρχείων είτε τοπικά είτε απομακρυσμένα. Για την υλοποίηση κάποιων παραμέτρων της εντολής (δείτε πιο κάτω) θα **ΠΡΕΠΕΙ** να ανακτήσετε και να αποθηκεύσετε

προσωρινά στον κατάλογο /tmp/\$USER τα δεδομένα, τα οποία θα επεξεργαστείτε στη συνέχεια με τις εντολές του UNIX που διδαχθήκατε στο μάθημα.

### Options:

#### (α) list-imap

Η επιλογή αυτή θα παρουσιάζει στην οθόνη μόνο τους καταλόγους του απομακρυσμένου IMAP server. Κανένα αρχείο δε θα κατεβαίνει τοπικά.

#### (β) show-body-message <message\_id>

Η επιλογή αυτή θα παρουσιάζει στην οθόνη μόνο το body του message με το δοθέν id.

#### (γ) show-subject-message <message\_id>

Η επιλογή αυτή θα παρουσιάζει στην οθόνη μόνο το subject του message με το δοθέν id.

#### (δ) find-string <string>

Η επιλογή αυτή θα κοιτάζει μέσα σε όλα τα μηνύματα που βρίσκονται στον κατάλογο /tmp/\$USER και θα παρουσιάζει στην οθόνη τις γραμμές των αρχείων που περιέχουν το string αναζήτησης. Αυτή η εντολή βέβαια προϋποθέτει ότι όλα τα αρχεία του IMAP έχουν ανακτηθεί στο /tmp/\$USER.

#### (ε) show-urls

Η επιλογή αυτή θα κάνει εξαγωγή όλων των email διευθύνσεων (π.χ., epl421@cs.ucy.ac.cy) από όλα τα μηνύματα που βρίσκονται μέσα στο λογαριασμό του χρήστη και θα τα παρουσιάζει στην οθόνη.

### V. Γενικοί Κανόνες

1. Το σύστημα δεν αφήνει ποτέ μεταβατικά αρχεία στον δίσκο, ούτε ανοικτά socket descriptors, ανεξάρτητα εάν διακοπεί η λειτουργία του προγράμματος από το κλείσιμο του κελύφους. Χρησιμοποιείτε την εντολή trap.
2. Το σύστημα πρέπει να χρησιμοποιεί τεχνικές δομημένου προγραμματισμού με την χρήση συναρτήσεων.
3. Το σύστημα πρέπει να ελαχιστοποιεί την χρήση πόρων του συστήματος (αρχεία, μνήμης, κτλ).
4. Το σύστημα πρέπει να μειώνει όσο το δυνατό περισσότερο τον χρόνο διεκπεραίωσης της ανάκτησης και επεξεργασίας των δεδομένων.

**Καλή Επιτυχία !**