ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

Τμήμα Πληροφορικής

ΕΠΛ 371 – Προγραμματισμός Συστημάτων

ΑΣΚΗΣΗ 4 – Ανάπτυξη Πολυνηματικού Εξυπηρετητή Αρχείων FTP

Διδάσκων: Δημήτρης Ζεϊναλιπούρ
Υπεύθυνος Εργασίας: Παύλος Αντωνίου

Ημερομηνία Ανάθεσης: Παρασκευή 28/03/2014
Ημερομηνία Παράδοσης: Τρίτη 15/04/2014 (19 μέρες)

(να υποβληθεί ηλεκτρονικά στο Moodle – η εξέταση της θα γίνει την τελευταία εβδομάδα)

I. Στόχος Άσκησης

Στόχος αυτής της εργασίας είναι η εξοικείωση με προχωρημένες τεχνικές προγραμματισμού διεργασιών, δια-διεργασιακής επικοινωνίας μέσω υποδοχών TCP/IP και πολυνηματικών εφαρμογών στη γλώσσα C. Ένας δεύτερος στόχος είναι να σας δοθεί η ευκαιρία να δουλέψετε ομαδικά για να υλοποιήσετε ένα ολοκληρωμένο σύστημα το οποίο θα κριθεί βάση της *ορθότητας*, *δομής* και *επίδοσής* του. Ένας τελευταίος στόχος είναι να σας δώσει την δυνατότητα να δουλέψετε με κάποια εργαλεία ανάπτυξης λογισμικού, όπως το Ολοκληρωμένο Περιβάλλον Ανάπτυξης Λογισμικού eClipse IDE for C/C++, το σύστημα εκδόσεων SVN και εργαλεία παραγωγής συνθετικού φόρτου για stress testing του συστήματος σας.

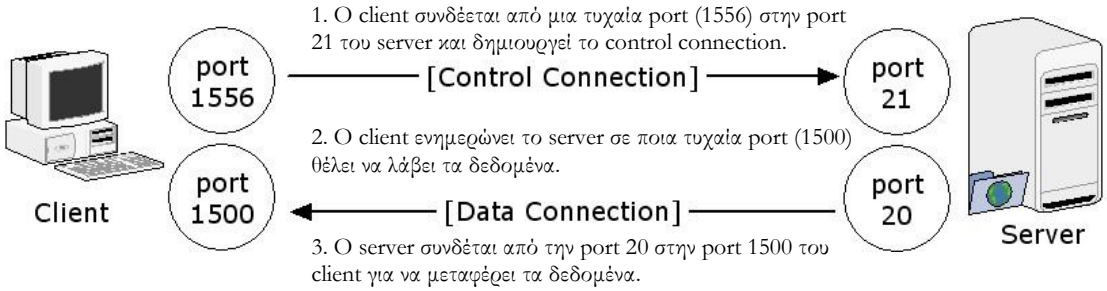
II. Αποστολή / Ανάκτηση αρχείων μέσω πρωτοκόλλου FTP

1. Τρόπος Λειτουργίας Πρωτοκόλλου

Το πρωτόκολλο FTP λειτουργεί με βάση το μοντέλο πελάτη-εξυπηρετητή (client-server) επιτρέποντας την αποστολή/ανάκτηση αρχείων μεταξύ των 2 αυτών οντοτήτων.
Αρχικά ο FTP server ανοίγει την θύρα (port) 21 περιμένοντας έναν FTP client να συνδεθεί. Στη συνέχεια ο client ξεκινά μια νέα σύνδεση από μια τυχαία θύρα προς την θύρα 21 του server. Μόλις γίνει η σύνδεση παραμένει ανοιχτή για όλη τη διάρκεια της συνόδου FTP. Η συγκεκριμένη σύνδεση ονομάζεται σύνδεση ελέγχου (**control connection**) και μέσα από το κανάλι αυτό στέλνονται όλες οι εντολές από τον client και οι απαντήσεις από το server, αλλά όχι δεδομένα (αρχεία). Έπεται η δημιουργία της σύνδεσης δεδομένων (**data connection**), της σύνδεσης με την οποία μεταφέρονται τα δεδομένα. Υπάρχουν δύο τρόποι για να δημιουργηθεί η σύνδεση δεδομένων μέσω της σύνδεσης ελέγχου, (α) με χρήση της ενεργητικής λειτουργίας (active mode) στην οποία τα δεδομένα σπρώχνονται (push) από τον server στον client ή (β) με χρήση της παθητικής λειτουργίας (passive mode) στην οποία τα δεδομένα αντλούνται (pull) από τον client.

A) Ενεργητική λειτουργία

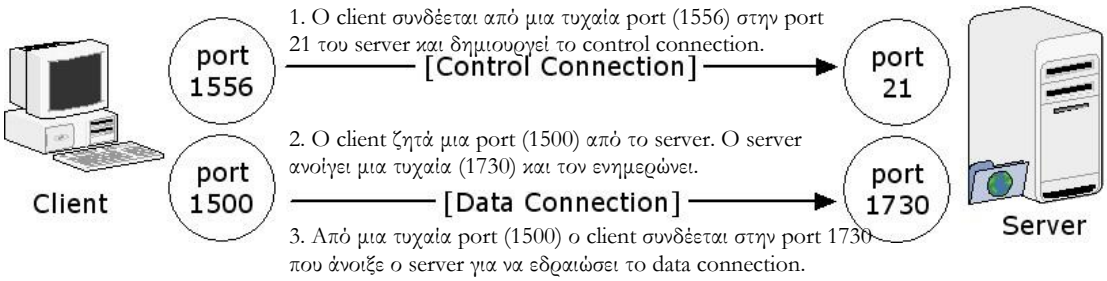
Στην ενεργητική λειτουργία (active mode) ο FTP client διαλέγει μια τυχαία θύρα στην οποία δέχεται τα δεδομένα της σύνδεσης. Ο client στέλνει τον αριθμό της θύρας, στην οποία επιθυμεί να "ακούει" (listen) για εισερχόμενες συνδέσεις. Ο FTP server δημιουργεί μια σύνδεση από την θύρα 20 στην ανοιχτή θύρα του client για τη μεταφορά των δεδομένων. Οποιαδήποτε πληροφορία ζητήσει ο client, ανταλλάσσεται με βάση αυτή τη σύνδεση, που βασίζεται στο TCP. Όταν η μεταφορά ολοκληρωθεί ο server κλείνει τη σύνδεση αποστέλλοντας ένα πακέτο FIN, όπως σε κάθε σύνδεση βασισμένη στο TCP. Κάθε φορά που ο client ζητάει δεδομένα, δημιουργείται κατά παρόμοιο τρόπο μια σύνδεση δεδομένων και η διαδικασία επαναλαμβάνεται.



Εικόνα 1: Διαγραμματική και διαλογική αναπαράσταση της ενεργητικής λειτουργίας FTP.

B) Παθητική λειτουργία

Στην παθητική λειτουργία (passive mode) ο client ζητά από τον server να διαλέξει μια τυχαία θύρα, στην οποία θα "ακούει" (listen) για την σύνδεση δεδομένων (data connection). Ο server ενημερώνει τον client για την θύρα την οποία έχει διαλέξει και ο client συνδέεται σε αυτή για τη μεταφορά των δεδομένων. Η μεταφορά ολοκληρώνεται όπως και στην ενεργητική λειτουργία (active mode), αφού η σύνδεση δεδομένων βασίζεται στο TCP.



Εικόνα 2: Διαγραμματική και διαλογική αναπαράσταση της παθητικής λειτουργίας FTP.

2. Εντολές FTP

Οι εντολές που περιγράφονται πιο κάτω είναι ένα υποσύνολο των εντολών που υποστηρίζει το πρωτόκολλο FTP, το οποίο δίνουμε απλά για κατανόηση του πρωτοκόλλου. **Στην ενότητα ΙΙΙ θα εξειδικεύσουμε τις εντολές τις οποίες θα πρέπει να υλοποιήσετε στα πλαίσια της παρούσας άσκησης.**

Εντολές ελέγχου πρόσβασης

USER <username>	καθορισμός κωδικού που είναι καταχωρημένος στο server	
PASS <password>	καθορισμός συνθηματικού που είναι καταχωρημένος στο server	
CWD <pathname>	αλλαγή καταλόγου	**(όπως cd)**
CDUP	μετάβαση στο «γονικό» κατάλογο	**(όπως cd ..)**

QUIT έξοδος

Εντολές προσδιορισμού παραμέτρων

PORT h1,h2,h3,h4,p1,p2 Δήλωση ενεργητικής λειτουργίας: κοινοποίηση τοπικής θύρας δεδομένων.

PASV Δήλωση παθητικής λειτουργίας: αίτηση για αποστολή θύρας αφούγκρασης (listening) του server για παραλαβή δεδομένων

TYPE καθορισμός τύπου δεδομένων: ascii (για κείμενο), image (για εικόνες και binary αρχεία), default τιμή: ascii

Εντολές υπηρεσίας

RETR <pathname> ανάκτηση (retrieve) αρχείου. Η εντολή αυτή μπορεί να εκτελεστεί αν υπάρχει ανοικτή σύνδεση δεδομένων (δηλαδή μετά από την εντολή PORT ή PASV). **(όπως cp remote_file local_file)**

STOR <pathname> αποστολή (store) αρχείου. Η εντολή αυτή μπορεί να εκτελεστεί αν υπάρχει ανοικτή σύνδεση δεδομένων (δηλάδη μετά από την εντολή PORT ή PASV). **(όπως cp local_file remote_file)**

APPE <pathname> αποστολή αρχείου και προσάρτηση σε υφιστάμενο αρχείο που βρίσκεται στο server. Η εντολή αυτή μπορεί να εκτελεστεί αν υπάρχει ανοικτή σύνδεση δεδομένων (δηλαδή μετά από την εντολή PORT ή PASV)

(όπως cat local_file >> remote_file)

ABOR διακοπή (abort) προηγούμενης εντολής υπηρεσίας

PWD εμφάνιση τρέχοντος καταλόγου **(pwd)**

LIST μεταφορά της λίστας των αρχείων **(ls -la)** Η εντολή αυτή μπορεί να εκτελεστεί αν υπάρχει ανοικτή σύνδεση δεδομένων (δηλαδή μετά από την εντολή PORT ή PASV). Η εντολή αυτή επιστρέφει παρόμοια αποτελέσματα με την εντολή ls –la. Οι καταλόγοι επισημαίνονται με "d".

DELE <pathname> διαγραφή αρχείου **(rm)**

MKD <pathname> δημιουργία καταλόγου **(mkdir)**

RMD <pathname> διαγραφή καταλόγου **(rmdir)**

SIZE <pathname> μέγεθος αρχείου (οκτάδες, 8-bit byte) σαν δεκαδικός αριθμός

STAT <pathname> εάν δοθεί όρισμα ένα directory τότε είναι το ίδιο με την εντολή list με τη διαφορά ότι η απάντηση από το server έρχεται μέσα από τη σύνδεση ελέγχου (χωρίς να πρέπει να ανοίξει σύνδεση δεδομένων)

Περισσότερες πληροφορίες μπορείτε να βρείτε στο RFC959 (http://www.ietf.org/rfc/rfc0959.txt) ή στον πιο κάτω σύνδεσμο: http://www.nsftools.com/tips/RawFTP.htm

III. Περιγραφή Ζητούμενων Εργασίας

Αντικείμενο της άσκησης είναι να αναπτύξετε ένα «πολυνηματικό» εξυπηρετητή αρχείων (ftp server), ο οποίος θα υποστηρίζει ένα βασικό υποσύνολο των εντολών του πρωτοκόλλου FTP. Η λειτουργία του πρωτοκόλλου FTP περιγράφεται στο τεχνικό άρθρο Request For Comments 959: http://tools.ietf.org/html/rfc959.

Για τις ανάγκες της άσκησης, πρέπει να μπορείτε να χειριστείτε τις εντολές **USER**, **PASS** **CWD, CDUP** και **QUIT** [RFC959/§4.1.1], την εντολή **PASV** [RFC1939/§4.1.2] καθώς και τις εντολές **RETR, STOR, LIST, MKD**, και **STAT** [RFC1939/§4.1.3]. Αν δοθεί κάποια από αυτές τις εντολές ο εξυπηρετητής πρέπει να απαντά με κάποιο θετικό κωδικό.

Αν δοθεί κάποια διαφορετική εντολή από τον πελάτη, τότε ο εξυπηρετητής πρέπει να απαντά με κάποιο αρνητικό κωδικό. Οι κωδικοί αυτοί θα αναφερθούν πιο κάτω αλλά περιγράφονται πιο αναλυτικά στο RFC959/§4.2.1 και §4.2.2.

Αρχικά, ο εξυπηρετητή FTP ξεκινά δημιουργώντας ένα TCP socket στη θύρα 21. Όταν πελάτης FTP (π.χ. WinSCP, Filezilla, κτλ.) επιθυμεί να στείλει ή να λάβει ένα αρχείο, δημιουργεί μια σύνδεση TCP με τον εξυπηρετητή FTP. Όταν θα εγκαθιδρυθεί η σύνδεση στη θύρα 21, ο εξυπηρετητής FTP στέλνει ένα χαιρετισμό. Αυτή η σύνδεση ονομάζεται σύνδεση ελέγχου. Στη συνέχεια, ο πελάτης και ο εξυπηρετητής FTP ανταλλάζουν εντολές και απαντήσεις αντίστοιχα μέσω της σύνδεσης ελέγχου. Αν ο πελάτης αποφασίσει να λάβει ή να στείλει ένα αρχείο από και προς τον εξυπηρετητή, αιτείται τη δημιουργία μιας σύνδεσης δεδομένων. **Στην άσκηση αυτή, καλείστε να υλοποιήσετε την παθητική λειτουργία όταν θα δημιουργείται μια σύνδεση δεδομένων**.

Η κάθε σύνδεση δεδομένων χρησιμοποιείται για τη μεταφορά ενός αρχείου και μετά καταργείται. Η επικοινωνία πελάτη-εξυπηρετητή τερματίζεται με την κατάργηση της σύνδεσης ελέγχου, από την πλευρά του πελάτη.
Οι εντολές FTP αποτελούνται από μια case-insensitive λέξη-κλειδί (command), πιθανώς ακολουθούμενη από κανένα ή ένα όρισμα. Όλες οι εντολές στέλνονται μέσα από τη σύνδεση ελέγχου. Όλες οι εντολές τερματίζονται από την ακολουθία "\r\n", δηλαδή από δυο χαρακτήρες, τον "\r" (Carriage Return) με ASCII κωδικό 13 και τον "\n" (Line Feed) με ASCII κωδικό 10. Μπορείτε να το δείτε και γραμμένο σαν CRLF. Οι λέξεις-κλειδιά και τα ορίσματα αποτελούνται από εκτυπώσιμους ASCII χαρακτήρες και διαχωρίζονται μεταξύ τους από ένα μόνο SPACE χαρακτήρα. Οι λέξεις-κλειδιά αποτελούνται από τρεις ή τέσσερις χαρακτήρες.

Οι λέξεις-κλειδιά μπορούν να χωριστούν σε: ελέγχου πρόσβασης, προσδιορισμού παραμέτρων ή αιτήσεις υπηρεσίας FTP, όπως δόθηκαν στην ενότητα ΙΙ. **Μερικές λέξεις-κλειδιά (όπως STAT, QUIT) μπορούν να σταλούν μέσα από τη σύνδεση ελέγχου ενόσω μεταφορά δεδομένων βρίσκεται σε εξέλιξη.**
Οι απαντήσεις στο πρωτόκολλο FTP αποστέλλονται (μέσω της σύνδεσης ελέγχου) για να διασφαλιστεί ο συγχρονισμός των αιτήσεων του πελάτη FTP και των ενεργειών του εξυπηρετητή FTP κατά τη διάρκεια της μετάδοσης ενός αρχείου, και για να υπάρχει εγγύηση ότι ο πελάτης γνωρίζει πάντα την κατάσταση του εξυπηρετητή. Κάθε εντολή πρέπει να δημιουργήσει τουλάχιστον μια απάντηση, αν και μπορεί να υπάρχουν περισσότερες από μία. Στην τελευταία αυτή περίπτωση, οι πολλαπλές απαντήσεις, πρέπει να διακρίνονται εύκολα. Επιπλέον, κάποιες εντολές εμφανίζονται σε διαδοχικές ομάδες, όπως οι εντολές USER και PASS. Μια αποτυχία σε οποιοδήποτε σημείο της ακολουθίας απαιτεί την επανάληψη ολόκληρης της ακολουθίας από την αρχή. Μια απάντηση FTP αποτελείται από έναν τριψήφιο αριθμό που ακολουθείται από κάποιο κείμενο. Μεταξύ του τριψήφιου αριθμού και του κειμένου μεσολαβεί SPACE ενώ η απάντηση τερματίζεται από το CRLF.

Η υλοποίησή σας θα πρέπει να επιστρέφει **τουλάχιστο** τους ακόλουθους κωδικούς απάντησης (ανάλογα με την εντολή του πελάτη):

150 File status okay; about to open data connection. [Ακολουθεί την εντολή RETR]
200 Command okay. [Ακολουθεί σωστή εκτέλεση μιας εντολής όπως CWD, CDUP, LIST]
220 Service ready for new user. [Επιστρέφεται όταν ο πελάτης συνδεθεί επιτυχώς με τον εξυπηρετητή]
221 Service closing control connection. Logged out if appropriate.
226 Closing data connection. Requested file action successful. [Ακολουθεί και επιβεβαιώνει τη σωστή αποστολή ή λήψη αρχείου από το data connection]

227 Entering Passive Mode (h1,h2,h3,h4,p1,p2). [Ακολουθεί την εντολή PASV]
230 User logged in, proceed. [Ακολουθεί την εντολή PASS όταν αυτή είναι επιτυχής]
250 Requested file action okay, completed. [Ακολουθεί σωστή εκτέλεση της εντολής STOR]
257 "PATHNAME" created. [Ακολουθεί την εντολή MKD]
331 User name okay, need password. [Ακολουθεί την εντολή USER όταν αυτή είναι επιτυχής]
332 Need account for login. [Όταν το USER σταλεί χωρίς username]
426 Connection closed; transfer aborted. [Όταν διακόπτεται η μετάδοση δεδομένων απότομα]
451 Requested action aborted: local error in processing. [Μήνυμα λάθους για πολλές εντολές όπως CWD, CDUP, MKD, STOR, PASV, LIST, STAT κ.α.]
500 Syntax error, command unrecognized. [Όταν ή εντολή είναι λεκτικά ορθή αλλά συντακτικά λάθος π.χ. filename STOR αντί ανάποδα]
502 Command not implemented. [Όταν η εντολή που δίδεται δεν έχει υλοποιηθεί π.χ. STAR αντί STOR]
503 Bad sequence of commands. [Όταν δοθεί λανθασμένη εντολή που δεν αναμένεται π.χ. μετά το USER να δοθεί STOR αντί PASS]
530 Not logged in. [Όταν δίνονται εντολές πριν δοθεί το USER]
550 Requested action not taken. File unavailable. [Όταν δεν υπάρχει το ζητούμενο αρχείο π.χ. μετά την εντολή RETR]

IV. Παράδειγμα Συνόδου FTP

Στο πιο κάτω παράδειγμα, ο πελάτης U θέλει να μεταφέρει αρχεία από/προς τον εξυπηρετητή S. Ένα τυπικό παράδειγμα τέτοιου σεναρίου φαίνεται πιο κάτω. Οι εντολές που στέλλονται από τον πελάτη δίδονται με το '---->' ενώ οι απαντήσεις του εξυπηρετητή με το '<---'.

```
Connect to host S, port L, establishing control connection.
<---- 220 Service ready for new user<CRLF>.
USER Doe<CRLF>---->
<---- 331 User name ok, need password <CRLF>.
PASS mumble<CRLF>---->
<---- 230 User logged in, proceed<CRLF>.

Client-FTP initiates PASV operation.
PASV<CRLF>---->
<----         227         Entering         Passive         Mode
(208,75,230,189,144,129)<CRLF>.
Connect to host S, port L2, establishing data connection.
Client-FTP opens local file in ASCII.
RETR test.pl1<CRLF> ---->
<---- 150 Accepted data connection<CRLF>.
Server sends data through data connection.
<----   226   File   status   okay;   about   to   open   data
connection<CRLF>.
Server closes data connection.

Connect to host S, port L2, establishing data connection.
Client-FTP opens local file in ASCII.
STOR test2.txt<CRLF> ---->
<---- 550 Requested action not taken. File unavailable<CRLF>
QUIT <CRLF> ---->
Server closes all connections.
```

V. Πολυνηματική λειτουργία εξυπηρετητή

Ο ftp server που θα υλοποιήσετε πρέπει να είναι σε θέση να εξυπηρετεί «ταυτόχρονα» πολλές αιτήσεις από πελάτες. Δεν πρέπει, δηλαδή, να τελειώσει πρώτα με την εξυπηρέτηση μιας αίτησης και μετά να δέχεται νέες. Για να το πετύχετε αυτό, θα πρέπει να εκμεταλλευτείτε τη δυνατότητα ύπαρξης πολλών *νημάτων* μέσα στη διεργασία του ftp server. Μια ιδέα θα ήταν, όταν παίρνει μια αίτηση από πελάτη, να δημιουργεί ένα νήμα για να την εξυπηρετήσει, ενώ το αρχικό νήμα να περιμένει νέες αιτήσεις. Η εξυπηρέτηση των αιτήσεων αυτών θα γίνεται από νέα νήματα που θα δημιουργεί το αρχικό. Φυσικά, όταν ένα νήμα τελειώνει την αποστολή του, θα πρέπει να τερματίζει.

Η πιο πάνω προσέγγιση δεν είναι πολύ καλή, γιατί δεν είναι ιδιαίτερα ελεγχόμενη η δημιουργία και καταστροφή νημάτων στην εφαρμογή, κάτι που μπορεί να αποβεί εξαιρετικά προβληματικό σε κάποιες περιπτώσεις.
Μια άλλη, καλύτερη, ιδέα είναι το αρχικό νήμα να δημιουργήσει ένα thread-pool, δηλαδή να δημιουργήσει εξ αρχής ένα σταθερό αριθμό νημάτων-εργατών (που το πλήθος τους να δίνεται) και όταν υπάρχει αίτηση για εξυπηρέτηση να την αναθέτει σε κάποιο από τα νήματα αυτά που δεν έχει δουλειά. Τα νήματα, αφού εξυπηρετήσουν ένα πελάτη, δεν τερματίζουν, αλλά μεταβαίνουν σε κατάσταση αναμονής. Φυσικά, αν δεν υπάρχει διαθέσιμο νήμα, το αρχικό θα πρέπει να περιμένει μέχρι να υπάρξει, χωρίς να δέχεται νέες αιτήσεις. Συγκεκριμένα εάν υπάρξουν περισσότερες αιτήσεις από το μέγιστο αριθμό νημάτων στο pool τότε το σύστημα απορρίπτει την αίτηση κλείνοντας το socket (χωρίς να επιστρέφει οποιανδήποτε απάντηση στον πελάτη). Άλλες παραμέτρους που χρειάζεται να πάρει ο ftp server σας, εκτός από το πλήθος των νημάτων, είναι ο αριθμός θύρας στον οποίο θα αναμένει αιτήσεις, ο κατάλογος-ρίζα του ιεραρχικού συστήματος αρχείων που «σερβίρει» και άλλες παραμέτρους που τυχόν χρησιμοποιήσετε. Οι παράμετροι μπορεί είτε να δίδονται ως ορίσματα στο πρόγραμμά σας ή καλύτερα να βρίσκονται σε κάποιο αρχείο config.txt, το οποίο θα έχει τη δομή

```
# Ftp Server Configuration File

# The Number of Threads in the Threadpool
THREADS=40

# The Port number of the ftp server
PORT=30000
…
```

Για να πειραματίζεστε με το πρόγραμμα που θα γράψετε, κατά τη φάση της ανάπτυξής του, μπορείτε να του στέλνετε αιτήσεις μέσω της εντολής telnet ("man telnet" για να δείτε πώς).

Για να δοκιμάσετε τον ftp server σας μπορείτε να δημιουργήσετε ένα δικό σας κατάλογο και να τοποθετήσετε εκεί τα αρχεία της αρεσκείας σας.

VI. Ανάπτυξη Λογισμικού

Ομάδες

Η άσκηση αυτή θα υλοποιηθεί σε ομάδες όπως εχει αναρτηθει στο Moodle, τα οποία αναμένεται να συμβάλουν ισομερώς σε χρόνο και ουσιαστική δουλειά.

Συντήρηση Εκδόσεων με το SVN

Για να γίνει καλύτερος συντονισμός μεταξύ των μελών της ομάδας σας αλλά και για να μπορέσουμε να παρακολουθήσουμε τη πρόοδο ανάπτυξης του λογισμικού σας (δηλ., πως ακριβώς έχει συμβάλει κάποιο άτομο στη διεκπεραίωση της εργασίας), **πρέπει** να γίνει χρήση του Συστήματος Έλεγχου Εκδόσεων ((Re)Version Control System) **Subversion (ή SVN)**. Παρακαλώ ανατρέξετε στο ΕΠΛ132 για περισσότερες πληροφορίες για το SVN.

VII. Αξιολόγηση

A) Τι πρέπει να παραδώσετε;

- **Στο Moodle:** Ενα αρχείο **ftp.tar.gz** το οποίο θα περιέχει:
  1. Τον πηγαίο κώδικα μαζί με το σχετικό Makefile,
  2. Ένα README.txt αρχείο οποίο θα δίδει οδηγίες χρήσης του συστήματός σας (περίπου 1 σελίδα) και
  3. Architecture (DOC ή PDF), το οποίο θα περιγράφει την αρχιτεκτονική του συστήματος, τις βασικές επιλογές στο σχεδιασμό αυτής της αρχιτεκτονικής, περιγραφή της επιπλέον λειτουργίας που αποφασίσατε να υλοποιήσετε, διάφορες δυσκολίες που αντιμετωπίσατε (~2-3 σελίδες).
- **Στο Εργαστήριο (κατά την μέρα της εξέτασης): Να παραδοθούν εκτυπωμένα τα 1-3.**

B) Κριτήρια Αξιολόγησης.

1. **30% - Δομή Συστήματος:** Το σύστημα πρέπει να χρησιμοποιεί τεχνικές δομημένου προγραμματισμού με τη χρήση συναρτήσεων, αρχείων επικεφαλίδας (.h), πολλαπλών αρχείων για καλύτερη δομή του πηγαίου κώδικα, Makefile, αρχείων ελέγχου (unit tests) τα οποία θα ελέγχουν την ορθότητα των συστατικών (modules) του συστήματος σας ανεξάρτητα από την υπόλοιπη λειτουργία του συστήματος, διαχείριση λαθών συστήματος με την perror, έλεγχος ταυτοχρονίας νημάτων με χρήση σηματοφόρων, κτλ.

2. **70% - Ορθότητα Λειτουργίας:** Το σύστημα θα πρέπει να διεκπεραιώνει ορθά τις λειτουργίες του συστήματος όπως αυτές περιγράφονται σε αυτή την εκφώνηση και το RFC959. Η εκφώνηση της άσκησης δεν σας δεσμεύει για τις δυνατότητες που θα έχει ο εξυπηρετητής που θα υλοποιήσετε. **Η εκφώνηση απλά θέτει ένα ελάχιστο όριο δυνατοτήτων που θα πρέπει να υλοποιήσετε.** Αυτό είναι σκόπιμο για να σας αφήσει αρκετή ελευθερία στη λήψη πρωτοβουλιών και στην εκδήλωση δημιουργικότητας από την πλευρά σας. Μέσα από αυτή την άσκηση θέλουμε να σας δοθεί η δυνατότητα να επεξεργαστείτε από μόνοι σας ένα τεχνικό έγγραφο (RFC959) καθώς επίσης να ανακαλύψετε νέες συναρτήσεις πέρα από αυτές που διδαχθήκατε ήδη στις διαλέξεις.

Καλή Επιτυχία !