

# Εργασία 1<sup>η</sup> 2018

## Ενσωματωμένα Συστήματα Πραγματικού Χρόνου

### Θεοδωρίδης Ξενοφών 8054

---

Link κώδικα: <https://drive.google.com/open?id=15uzhWdDC57znZOlu6tkS3-cpWolifJFN>

#### Εισαγωγή

Σε αυτή την εργασία μας ζητήθηκε να υλοποιήσουμε τις εκδόσεις με pthreads και διεργασίες ώστε ο server να μπορεί να εξυπηρετεί ταυτόχρονα πολλούς clients.

#### Server.c

##### Υλοποίηση με pthreads:

Συνεχίζοντας το πρόγραμμα του κ.Φλώρου έπρεπε να γίνει μία τροποποίηση στον αλγόριθμο ώστε να είναι συμβατός με τη δομή των pthreads. Πιο συγκεκριμένα:

- ❖ Πριν τη συνάρτηση accept δημιουργείται αένας βρόγχος.
- ❖ Ελέγχω αν η συνάρτηση accept απέτυχε να εκτελεστεί και εμφανίζω μήνυμα λάθους.
- ❖ Σε περίπτωση επιτυχίας δημιουργώ ένα καινούργιο thread και με τη συνάρτηση connection\_handler ολοκληρώνω τη διαδικασία.
- ❖ Αναλυτικότερα η συνάρτηση connection\_handler ελέγχει με βάση το αποτέλεσμα της συνάρτησης read και τυπώνει είτε το μήνυμα που στάλθηκε , είτε μήνυμα αποσύνδεσης χρήστη είτε μήνυμα σφάλματος.
- ❖ Πριν το τέλος της συνάρτησης connection\_handler κλείνω το socket τη διεύθυνση του οποίου πέρασα σαν όρισμα.

### Υλοποίηση με διεργασίες:

Δεν διαφέρει και πολύ από την υλοποίηση με pthreads. Το μόνο που αλλάζει είναι ότι στον αέναο βρόγχο γίνεται χρήση της fork(). Το mother process είναι ο αέναος βρόγχος και το child process καλεί την συνάρτηση connection\_handler ( ίδια με αυτή στην υλοποίηση των pthreads απλά αυτή τη φορά δέχεται όρισμα int \* και όχι void \* όπως απαιτείται από την pthread\_create).

Αναφέρω ότι τόσο στην υλοποίηση με τα threads τόσο και στην υλοποίηση με τις διεργασίες δεν χρειάστηκε να γίνει pthread\_join και waitpid ( με κάποιο status) καθώς η εξυπηρέτηση του κάθε client είναι πλήρως ανεξάρτητη από την εξυπηρέτηση κάθε άλλου.

### Client.c

Δεν έγινε καμία τροποποίηση πάνω στον αλγόριθμο του κ.Φλώρου. Το δοκίμασα και γινόταν επιτυχής σύνδεση τόσο από δύο διαφορετικούς Η/Υ όσο και από το OpenWrt ( που έτρεχε το server.c ) με τον προσωπικό μου Η/Υ.