



## ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

### ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2024-2025, ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ

ΜΑΘΗΜΑ: ΠΡΩΤΟΚΟΛΛΑ ΔΙΑΔΙΚΤΥΟΥ

4<sup>η</sup> ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

Καταληκτική ημερομηνία υποβολής των απαντήσεων: **30/04/2024 23:55**.

Ε. Χαλεπλίδης ([ehalep@unipi.gr](mailto:ehalep@unipi.gr))

Α. Ρούσκας ([arouskas@unipi.gr](mailto:arouskas@unipi.gr))

Σε συνέχεια της δεύτερης, τρίτης και τέταρτης εργαστηριακής διάλεξης, καλείστε να δημιουργήσετε και να υλοποιήσετε ένα πρωτόκολλο επικοινωνίας. Έχετε απόλυτη ευχέρεια στην δημιουργία των μηνυμάτων του πρωτοκόλλου με μόνη προϋπόθεση να σχεδιάσετε το κατάλληλο πρωτόκολλο (με τους αντίστοιχους headers) ώστε να ικανοποιεί τις απαιτήσεις της άσκησης που θα επεξηγηθεί παρακάτω.

**Σημείωση:** Δεν υπάρχει σωστός ή λάθος σχεδιασμός σε headers, και δεν ζητείται να είναι αποδοτικό το πρωτόκολλό σας.

Τα παραδοτέα της εργαστηριακής άσκησης είναι:

1. Εκτελέσιμος κώδικας για τον server
2. Εκτελέσιμος κώδικας για τον client
3. Παράδοση τεκμηρίωσης με μορφής IETF draft

Για την μορφή IETF draft, θα πρέπει να χρησιμοποιήσετε τα εργαλεία της IETF:

1. IETF author tools – <https://author-tools.ietf.org/> – Για την μετατροπή από XML σε TXT.
2. idnits – <https://www6.ietf.org/tools/idnits> – Για τον βασικό έλεγχο του draft ότι πληρεί όλες τις προδιαγραφές.
3. idspell – <https://tools.ietf.org/tools/idspell/idspell.pyht> – Για τον ορθογραφικό έλεγχο του κειμένου.

Θα πρέπει να καταγράψετε την δομή και την αλληλουχία των μηνυμάτων και να επεξηγήσετε τα πεδία των μηνυμάτων, το Transport Protocol και την θύρα που χρησιμοποιήσατε. **Προσοχή:** Δεν καταγράφουμε λεπτομέρειες υλοποίησης.

**Άσκηση:** Πρέπει να φτιάξετε ένα πρωτόκολλο επικοινωνίας μεταξύ ενός server και πολλαπλών client. Ο Server δρα σαν μία αναλυτική μηχανή η οποία εκτελεί τις εξής 3 πράξεις:

Α. Πολλαπλασιασμός N προσημασμένων ακεραίων, όπου N μεταξύ 2 και 10. Ο κάθε ακέραιος που θα δέχεται o Server πρέπει να είναι από -5 έως και 5. O server θα επιστρέφει το αποτέλεσμα του πολλαπλασιασμού.

Β. Μέσος όρος N ακεραίων, όπου N μεταξύ 2 και 20. Ο κάθε ακέραιος που θα δέχεται o Server πρέπει να είναι από 0 μέχρι 200. O Server θα επιστρέφει το αποτέλεσμα του μέσου όρου.

Γ. Αφαίρεση 2 σετ από N αριθμούς, όπου N μεταξύ 2 και 10. Ο κάθε ακέραιος που θα δέχεται o Server πρέπει να είναι από 0 έως και 60000. O Server θα δέχεται 2 σετ από N αριθμούς και θα αφαιρεί από κάθε αριθμό του πρώτου σετ με τον αντίστοιχο αριθμό του δεύτερου σετ. Θα επιστρέψει δηλαδή 1 σετ από N αριθμούς. Π.χ. αν o client στείλει τους αριθμούς στο 1<sup>ο</sup> σετ: 2, 4 και 6 και στο 2<sup>ο</sup> σετ 1, 3 και 4, o server θα επιστρέψει 1, 1 και 2.

Ο client θα πρέπει να στέλνει στον server όσους αριθμούς χρειάζεται και την εντολή που θέλει να εκτελεστεί, και θα πρέπει να λαμβάνει πίσω σαν απάντηση είτε το αποτέλεσμα αν είναι επιτυχημένη η πράξη ή μήνυμα λάθους (και τι είδος λάθους - αν π.χ. διθούν σαν είσοδο αριθμοί μεγαλύτεροι από το προαπαιτούμενο).

Στον κώδικα σας να εκτελεστούν και οι 3 πράξεις με όποια σειρά θέλετε, με προτίμηση διαβάζοντας τους αριθμούς από το πληκτρολόγιο για κάθε πράξη ή βάζοντάς τους τυχαία από κάποια συνάρτηση για τυχαίους αριθμούς.

O server θα πρέπει να μπορεί να υποστηρίξει πολλαπλούς **clients** με βάση την εργαστηριακή διάλεξη 3.

Στην ιστοσελίδα αυτή <https://docs.python.org/3/library/struct.html> μπορείτε να βρείτε όλους τους κωδικούς για την μορφοποίηση των δεδομένων στην παράγραφο **Format Characters**. Στο πεδίο **Standard Size**, σας αναφέρει τον αριθμό των bytes που χρησιμοποιεί κάθε τύπος δεδομένων.

Για να έχετε μία εικόνα για το εύρος των αριθμών μπορείτε να το υπολογίσετε με τον παρακάτω τύπο:

- Για n byte signed integers = (-2<sup>n\*8</sup>/2) έως (2<sup>n\*8</sup>/2 - 1)
- Για n byte unsigned integers = 0 έως (2<sup>n\*8</sup> - 1)

Συγκεκριμένα:

- 1 byte unsigned integer = 0 έως 255
- 2 byte unsigned integer = 0 έως 65535
- 4 byte unsigned integer = 0 έως 4294967295
- 1 byte signed integer = -128 έως 127
- 2 byte signed integer = -32768 έως 32767
- 4 byte signed integer = -2147483648 έως 2147483647

**Σημείωση:** Όταν κάνετε δοκιμές δοκιμάστε να βάλετε και σωστά νούμερα, αλλά και λανθασμένα για να εξετάσετε κάθε περίπτωση.

**Σημείωση:** Η βαθμολογία της άσκησης γίνεται με τον εξής υπολογισμό:

Όσον αφορά τον κώδικα:

- Αν καταθέσετε μόνο κώδικα για έναν client και έναν server (εργασία #2): 60%
- Αν καταθέσετε τον τελικό κώδικα για multi-client server (εργασία #3): 70%

Όσον αφορά το documentation:

- Αν καταθέσετε το τελικό τυπικό documentation (draft) σε txt (εργασία #4): 30%

Η εργασία είναι ατομική!

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!!!