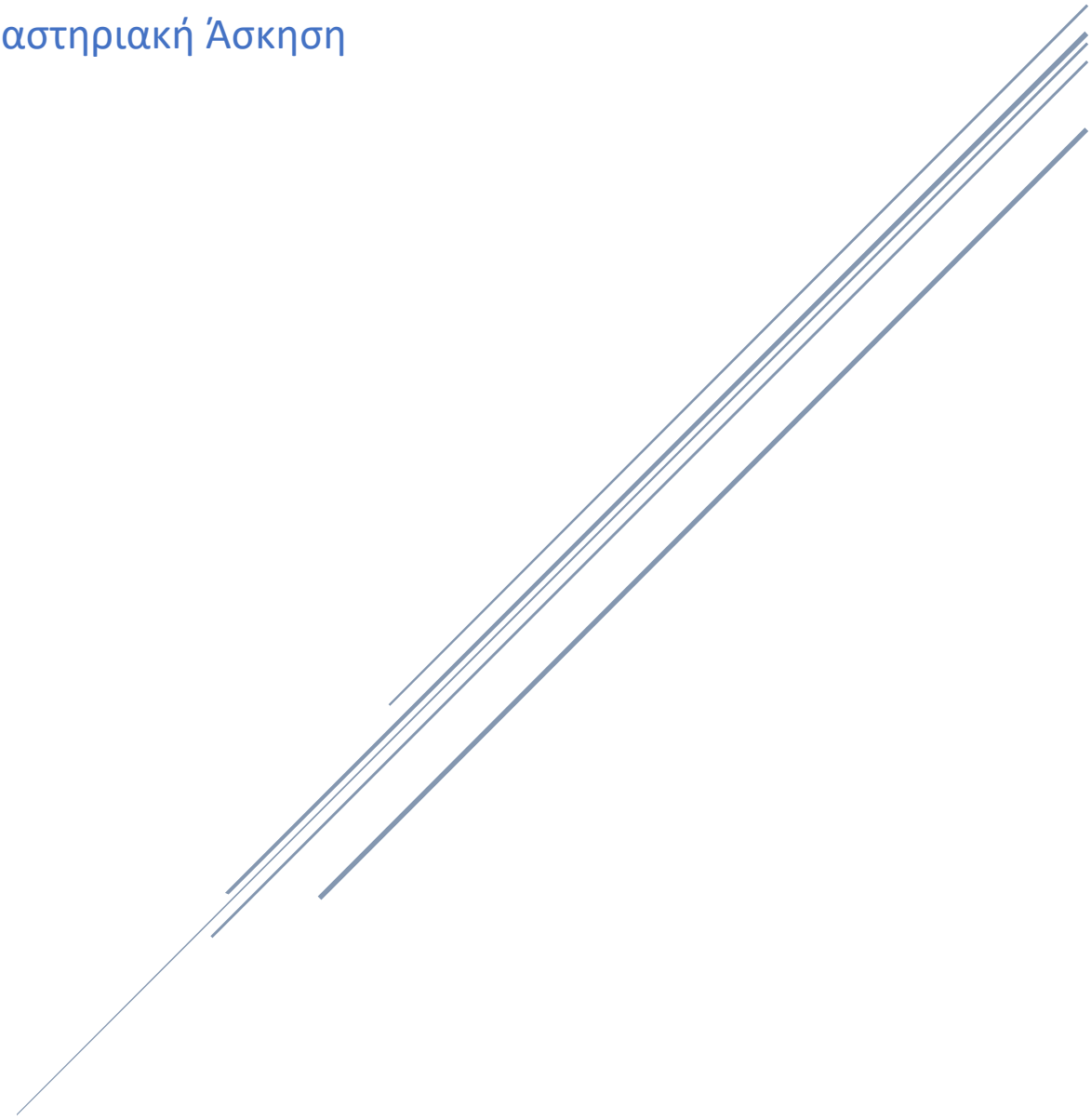
ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΥΣΤΗΜΑΤΩΝ

3η Εργαστηριακή Άσκηση



Μπουραΐμης Φώτης 2013030067 LAB21143241

12/01/2020

Πρόλογος:

Στην τελευταία εργαστηριακή άσκηση του μαθήματος αναλάβαμε να υλοποιήσουμε ένα σύστημα απομακρυσμένης εκτέλεσης εντολών. Το σύστημα αυτό αποτελείται από έναν εξυπηρετητή και έναν πελάτη και χρησιμοποιεί τα δύο πρωτόκολλα επικοινωνίας UDP και TCP για την μεταξύ τους επικοινωνία.

remoteClient:

Το πρόγραμμα remoteClient αναπαριστά τον πελάτη του συστήματος καθώς στέλνει εντολές στον server τις οποίες διαβάζει μέσα από ένα αρχείο που του υποδεικνύει ο χρήστης. Αφού το πρόγραμμα κάνει τα απαραίτητα για τη δημιουργία ενός TCP socket το οποίο συνδέεται στον εξυπηρετητή, κι ενός UDP socket το οποίο ακούει σε μια συγκεκριμένη θύρα (προκαθορισμένη πάλι από το χρήστη) για εισερχόμενα πακέτα, περνάμε σε δύο βασικές δομές της υλοποίησης.

Δομή Parent (remoteClient):

Στη δομη του πατέρα αναλαμβάνουμε να στείλουμε το αρχείο στον server. Συγκεκριμένα κάνουμε προσπέλαση του αρχείου γραμμή γραμμή, αποθηκεύουμε σε έναν buffer αυτή τη γραμμή αφού έχουμε φροντίσει να βάλουμε στην αρχή της το receivePORT έτσι ώστε να ξέρει ο server που πρέπει ν` απαντήσει κι έπειτα στέλνουμε το μήνυμα. Επίσης κάθε 10 εντολές σταματάμε να στέλνουμε μηνύματα για λίγα δευτερόλεπτα.

Δομή Child (remoteClient):

Σε αυτό το κομμάτι δεχόμαστε την απάντηση της κάθε εντολής. Αρχικά έχουμε φροντίσει με κατάλληλο string manipulation να δημιουργούμε το εκάστοτε όνομα όπως ζητείται στην εκφώνηση για την αποθήκευση των αποτελεσμάτων και έπειτα αποθηκεύουμε το εισερχόμενο μήνυμα στο αρχείο.

Στοιχεία Υλοποίησης (remoteClient):

Στη συνολική δομή του προγράμματος ήταν αναγκαίο να υπάρχει παραλληλισμός έτσι ώστε να μπορούμε ταυτόχρονα να δεχόμαστε αλλά και να στέλνουμε μηνύματα γι` αυτό επιλέξαμε να δημιουργήσουμε μια διεργασία παιδί. Ακόμη, για την σειρά των πακέτων που δεχόμαστε σε UDP ακολουθήσαμε τη σύμβαση με την οποία όλα τα πακέτα πρέπει να διαθέτουν το χαρακτηριστικό σύμβολο και το τελευταίο πακέτο κάθε εντολής ένα διαφορετικό σύμβολο έτσι ώστε να διαχειριζόμαστε πακέτα μεγαλύτερα από 512 bytes. Τέλος με την χρήση pipe ο πατέρας ενημερώνει το παιδί για το πόσα αρχεία πρέπει να περιμένει μέχρι να σταματήσει τη λειτουργία του.

remoteServer:

Το πρόγραμμα remoteServer διαχειρίζεται τις συνδέσεις των πελάτων, την εκτέλεση των εντολών και την αποστολή των αποτελεσμάτων στους πελάτες. Όπως και στο παραπάνω πρόγραμμα έτσι κι εδώ έχουμε διεργασίες παιδιά το πλήθος των οποίων εξαρτάται από τον χρήστη. Αρχικά δημιουργούμε τις διεργασίες και κάνουμε τα απαραίτητα για την αποδοχή συνδέσεων. Για την διαχείριση παραπάνω του ενός πελάτες, χρησιμοποιήσαμε τη select όπως περιγράφεται στην εκφώνηση και η επικοινωνία μεταξύ πατέρα παιδιων γίνεται αποκλειστικά με pipes. Τέλος για την εγκυρότητα των εντολών χρησιμοποιήσαμε regex.

Δομή Parent (remoteServer):

Ο πατέρας εφόσον δεχτεί μήνυμα από ήδη υπάρχοντα πελάτη ακολουθέι την εξής διαδικασία. Ελέγχει την εγκυρότητα της εντολής και επικοινωνεί μέσα από pipes το μέγεθος του μηνύματος, το ίδιο το μήνυμα, την

κατηγορία της εντολής όπως επίσης και το σημείο που βρίσκεται το receivePORT μέσα στο μήνυμα. Τα παραπάνω είναι η πληροφόρια η οποία διαθέτει ο πατέρας στα παιδια.

Δομή Child (remoteServer):

Το κάθε παιδί με τη σειρά του διαβάζει αυτήν την πληροφορία και έχει όλα τα απαραίτητα στοιχεία για να τρέξει την εντολή και να τη στείλει στον αντίστοιχο πελάτη. Εδώ να πούμε ότι το κάθε παιδί δε χρειάζεται να αξιολογήσει την εντολή που δέχεται (πχ. αν είναι έγκυρη η όχι), διότι πράττει ανάλογα με την κατηγορία της εντολής που του έχει δώσει ο πατέρας.

Στοιχεία Υλοποίησης (remoteServer):

Το πρόβλημα του συγρονισμού των παιδιών μας το λύνει το pipe το οποίο εμπεριέχει το μέγεθος του μηνύματος. Το κάθε παιδί αφου διαβάσει τον αριθμό αυτόν, ξέρει ακριβώς το μέγεθος σε bytes που πρέπει να περιμένει από το άλλο pipe.