

ΑΣΚΗΣΗ 1

ΕΡΓΑΣΤΗΡΙΟ ΔΙΚΤΥΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΧΡΙΣΤΟΠΟΥΛΟΣ ΑΛΕΞΑΝΔΡΟΣ 5928 ΕΤΟΣ 4

Serv1: Αρχικά εκτελείται ένας ατέρμων βρόχος στην κάθε επανάληψη του οποίου γίνεται `accept()` ώστε να συνδεθεί ο client εφόσον αρχικά χρησιμοποιείται το port του σερβερ.

ΛΕΙΤΟΥΡΓΙΑ: στην συνέχεια με `read()` διαβάζει ανά 200 bytes εντολών που έχει στείλει ο client έως ότου η `read()` επιστρέψει 0. Αρχικά κοιτάει το πρώτο byte: αν είναι `g` (103) διαβάζει τα επόμενα bytes τα οποία αποθηκεύει σε ένα προσωρινό string και αν συναντήσει `\0` σταματάει και εκτελεί την συνάρτηση `get` βάσει του αλφαριθμητικού που δημιουργήθηκε προηγουμένως, απαντά στον client και συνεχίζει στο επόμενο byte. Αν αυτό είναι `p` (`ascii p -> 112`) πρόκειται για μια εντολή `put` τα επόμενα bytes μέχρι το πρώτο αμέσως μετά `\0` τοποθετούνται σε ένα string `key` και τα αμέσως επόμενα μετά το `\0` έως το αμέσως επόμενο `\0` τοποθετούνται σε string `value` και στην συνέχεια γίνεται η κλήση της `put(key,value)`;

Για την αποθήκευση της δομής χρησιμοποιούνται δύο string arrays (`char ** key, char** value`) και ένα index το οποίο δείχνει το τελευταίο στοιχείο που προστέθηκε πρόσφατα.

Αν το byte είναι `@` (δηλαδή λάθος εντολή, χρησιμοποιήθηκε ο κωδικός 100 σε αυτήν την περίπτωση) κλείνουμε το socket επικοινωνίας και προχωρούμε στην επόμενη επανάληψη της `while(1)`.

Serv2: Σε αυτήν την περίπτωση εκτελείται ένας ατέρμων βρόχος. Εσωτερικά αν γίνει `accept()`

Εκτελείται `fork()` και στην συνέχεια το παιδί εκτελεί την λειτουργία που περιγράφηκε πιο πάνω. Ο πατέρας κλείνει το socket() του παιδιού.

Serv3: Αυτός ο σερβερ αρχικά διαβάζει την Τρίτη παράμετρο όπως αναφέρεται στην εκκώνηση (έστω έχει τιμή `x`) και στην συνέχεια εκτελεί `x` φορές `fork()` ώστε να δημιουργηθούν `x` διεργασίες. Κάθε παιδί έχει έναν ατέρμων βρόχο εσωτερικά του οποίου βρίσκεται πρώτη η `accept()` ώστε το παιδί να δεχθεί σύνδεση και στην συνέχεια αυτό το παιδί εκτελεί τις λειτουργίες όπως ο σερβερ 1. Αυτή η διαδικασία εκτελείται από το κάθε παιδί (`for(i=0;i<x;i++){ if(fork()==0){ while(1){ accept() treat_request() } } }`) (`x=argv[2]`).

Serv4: Η κυρίως λειτουργία εδώ είναι ένας ατέρμων βρόχος σε κάθε επανάληψη του οποίου αν γίνει `accept` στο socket `newsockfd` δημιουργείται ένα νήμα το οποίο χρησιμοποιεί το `newsockfd` και εξάγει την βασική λειτουργία που περιγράφεται στην αρχή πάνω σε αυτό.

Λειτουργία-συνάρτηση νήματος: δέχεται το socket που δημιουργήθηκε από την `accept` και εξάγει πάνω σε αυτό την λειτουργία πιο πάνω.

Για την αποφυγή race conditions χρησιμοποιείται ένα mutex το οποίο κλειδώνει κατά την εκτέλεση μίας `put` ή `get` ώστε αν εκτελούνται πολλά νήματα μαζί να μην υπάρχουν εξαρτήσεις δεδομένων.