

System programming – Project 3

Φωτιάδης Μιχαήλ – 1115201700183

Compile:
make

Run:

make run1	make valgrind1	./bin/whoServer ...
make run2	ή make valgrind2	ή ./bin/master ...
make run3	make valgrind3	./bin/whoClient ...

Notes:

Για τον whoServer:

Δημιουργώ τα threads και τα αποθηκεύω σε ένα thread pool, έπειτα ακούω στα 2 ports χρησιμοποιώντας pselect, όταν η pselect γυρίσει τότε αν είναι νέο connection, το κάνω accept και το βάζω να το παρακολουθεί η pselect, έτσι ο server δεν είναι ευάλωτος σε ddos attacks. Αν δεν είναι νέο connection το βάζω στον κυκλικό buffer και κανω signal τα threads πως υπάρχει δουλειά. Όπου χρειάζεται για να αποφευχθούν τα race conditions χρησιμοποιώ mutex και condition variables.

Τα threads αναμένουν μέχρι να υπάρχει δουλειά, τότε αναλαμβάνουν την διεκπεραίωση της. Ανάλογα απ το ποιο port προέρχεται ένα connection, το χειρίζομαι ανάλογα.

Αν προέρχεται απο το port των στατιστικών τότε είτε ένα worker στέλνει στατιστικά (τα οποία απλα τα κανω free()) ή προκειται για διαδικασία “Handshake” όπου ένας worker το port στο οποίο κάνει listen() και τις χώρες που διαχειρίζεται, και αποθηκεύω τις πληροφορίες αυτές.

Αν είναι query τότε το thread προωθεί το ερώτημα στον/στους κατάλληλους workers και αναμένει απάντηση την οποία και επιστρέφει με κατάλληλη μορφή. Το thread πριν προωθήση το query κάνει κάποιους τυπικούς ελέγχους και ένδεχεται να επιστρέψει κάποια errors στον client όπως

Error 400: Bad Request ή Error 500: Internal Server Error αν κάτι αποτύχει ή ακόμα Error 501: Country Not Registered.

Για τον whoClient:

Ξεκινάω να διβάζω γραμμή γραμμή το αρχείο μέχρι να διαβάσω έσως και numThreads γραμμές, για κάθε γραμμή που διαβάσω δημιουργώ και ένα thread και του περνάω το query.

Τα threads περιμένουν να τους κάνει signal ο πατέρας για να κάνουν connect και να στείλουν το query στον server.

Το main thread αφού δημιουργήσει όλα τα threads και τους κάνει signal τότε περιμένει να τερματίσουν και επαναλαμβάνει ολη την διαδικασία από την αρχή μέχρι να τελειώσει το αρχείο.

Για τον master:

Ίδιο με προηγούμενη εργασία απλά κάνει listen() σε ένα port, στέλνει τα στατιστικά στον server αντί για τον worker, και δέχεται queries και από τους δυο.

Συμαντική σημείωση !!!

1) Το πιο πιθανό είναι κανένας patient να μην κάνει exit γιατί θα πρέπει να έχει ίδιο id first name last name idease country age στα δεδομένα που παραδίδω μαζί με την εργασία.

2) Με την εκτέλεση της searchPatientRecord είναι πιθανό να υπάρχουν παραπάνω από ένας Patient με το ίδιο id. Αυτό ΔΕΝ είναι λάθος αλλά σχεδιαστική επιλογή! Κάθε worker κρατάει δικό του αρχείο με τα id, οπότε 2 patient θα μπορούσαν να έχουν ίδιο id αλλά σίγουρα θα έχουν διαφορετική χώρα.

Και τα 3 προγράμματα δεν έχουν leaks πουθενά