

Εργασία 3 Προγραμματισμός συστήματος

Φοιτητής: Κωνσταντίνος Λιακόπουλος 1115201600092

Εντολή μεταγλώττισης travelMonitorClient.c : gcc -o
travelMonitorClient travelMonitorClient.c citizen.c list.c travelFunctions.c
skipList.c

monitorServer.c : gcc -o monitorServer
monitorServer.c citizen.c list.c monitorFunctions.c skipList.c -lpthread

Εντολή εκτέλεσης: ./travelMonitorClient -m 5 -b 100 -c 5 -s 100 -l
input_dir -t 3

TravelMonitorClient.c:

Ξεκινώντας το πρόγραμμα δημιουργώ έναν πίνακα από
monitor[numMonitor] όπου θα αποθηκεύω τις πληροφορίες κάθε
monitor.

Περνάω τις πληροφορίες για το ποιες χώρες θα διαχειρίζεται κάθε
monitor , το port στο οποίο θα ακούει για συνδέσεις καθώς και ένα
πίνακα με τα ορίσματα που θα δωθεί στην exec. Η διαδικασία αυτή
γίνεται στις συναρτήσεις passingCountriesIntoChilds() και
passingArguments(). Στην συνέχεια καλεί την execn με ορίσματα
των πίνακα που έχω δημιουργήσει για το κάθε monitor.

Connect στα monitorServer: Αφού δημιουργηθούν τα monitorServer
(περιγράφω αναλυτικά παρακάτω) το TravelMonitorClient περιμένει
τα συνδεθεί (έχω βάλει μία sleep(3) έτσι ώστε να προλάβουν να
δημιουργηθούν οι servers) με την gethostbyname βρίσκω την public
IP του δικτύου μου και κάνω connect στο port του αντίστοιχου
monitorServer. Στην συνέχεια είναι έτοιμο να δεχθεί ερωτήματα.
(Δεν έχω υλοποιήσει το κομμάτι με τα bloom Filters καθώς είχα
διάφορες δυσκολίες)

monitorServer.c

Το πρώτο πράγμα που κάνω είναι να διαβάσω τα αρχεία των χωρών
από τα path που μου έχει στείλει το travelMonitor μέσω threads από
ένα κυκλικό buffer. Βασίστηκα στο παράδειγμα του κ.Ντούλα με
των producer/consumer αλλά υλοποιώντας το με πολλούς
consumers. Για να γίνει αυτό έχω δημιουργήσει ένα extra condition

var και ένα extra από αυτά του παραδείγματος του κ.Ντούλα. Ο producer ξεκινάει να γεμίζει το κυκλικό buffer και παράλληλα στέλνει signal για να τα διαβάζουν οι consumers. Σε περίπτωση που γεμίσει περιμένει να διαβαστούν κάποια αρχεία μέχρι να βάλει τα υπόλοιπα. Μόλις κάποιο thread διαβάσει και το τελευταίο αρχείο στέλνει signal και κάνει unlock to mutex για να τερματίσουν και τα υπόλοιπα threads.

Δημιουργία Server: Μόλις τερματίσουν επιτυχώς όλα τα threads (με χρήση της join) δημιουργώ τον server που θα συνδεθεί το travelMonitorClient μόλις συνδεθεί κάνω fork τερματίζει το parent process και το παιδί είναι έτοιμο να δεχτεί ερωτήματα από το travelMonitorClient. Για την υλοποίηση των ερωτημάτων χρησιμοποιώ την ίδια λογική με την εργασία 2 με την διαφορά ότι η επικοινωνία γίνεται πάνω από socket και δεν κάνω trigger κάποιο monitor με κάποιο signal αλλά η πληροφορία στέλνεται σε όλα τα monitorServer και απαντάει το κατάλληλο