ΕΡΓΑΣΙΑ #3 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ
Group : ΑΡΤΙΟΙ

Α.Μ. : 1115201200106
Όνομα : Βασίλειος
Επίθετο : Μαυρομμάτης

ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ

Η εργασία είναι υλοποιημένη σε mix C/C++/Unix Syscalls.
Όλες οι δομές που έχουν υλοποιηθεί είναι δικές μου από εργασίες μαθημάτων κυρίως Αντικειμενοστραφή Προγραμματισμού, Λειτουργικών Συστημάτων και Δομών Δεδομένων, καθώς και των προηγούμενων δύο εργασιών Προγραμματισμού Συστήματος με μικροαλλαγές για βελτιστοποίηση στο ζητούμενο.
Η εργασία αποτελείται από τα εξής **8** αρχεία:
1 αρχείο **Dropbox_server.cpp** το οποίο περιέχει την main του προγράμματος server.
1 αρχείο **Dropbox_client.cpp** το οποίο περιέχει την main του προγράμματος client.
2 αρχεία **HelpersClient.cpp** και το αντίστοιχο header **HelpersClient.hpp** που έχουν βοηθητικές δομές και συναρτήσεις ελέγχου εγκυρότητας δεδομένων τόσο για το input του χρήστη όσο και για τον έλεγχο των clients που εισέρχονται ή φεύγουν απ'το σύστημα, καθώς και δομές ελέγχου εκδόσεων αρχείων, και χρησιμοποιούνται απ'το dropbox_client.
2 αρχεία **HelpersServer.cpp** και το αντίστοιχο header **HelpersServer.hpp** που έχουν βοηθητικές δομές και συναρτήσεις ελέγχου εγκυρότητας δεδομένων τόσο για το input του χρήστη όσο και για τον έλεγχο των clients που εισέρχονται ή φεύγουν απ'το σύστημα και χρησιμοποιούνται απ'το dropbox_server.
1 αρχείο bash script **create_files_small.sh** το οποίο δημιουργεί τα input directories και subdirectories για τον κάθε client μέχρι το ζητούμενο επίπεδο και τα γεμίζει με αρχεία μεγέθους ~ 1KB κατανεμημένα στα directories με χρήση Round-Robin.
1 αρχείο **Makefile** για την δημιουργία των **2** τελικών εκτελέσιμων προγραμμάτων : **dropbox_client** και **dropbox_server**.

1)ΛΕΙΤΟΥΡΓΙΑ SERVER:

Αρχικά η main του προγράμματος ελέγχει οτι τα αρχικά arguments που απαιτούνται (και μπορεί να έρθουν με οποιδήποτε σειρά) είναι έγκυρα και αν ναι προχωράει.
Έπειτα αφού δημιουργήσει το socket μπαίνει σε έναν ατέρμονο βρόγχο όπου και εξυπηρετεί σειριακά εισερχόμενες συνδέσεις από τα προγράμματα client ελέγχωντας πάντα για την εγκυρότητα των αιτημάτων , καθώς και ενημερώνοντας τις όποιες τοπικές δομές ελέγχου χρησιμοποιέι.
Τα αιτήματα που εξυπηρετεί ο server είναι τα εξής τρία: 1) LOG_ON <IP address, portNum> , 2) GET_CLIENTS και 3) LOG_OFF.
Τέλος ο server τερματίζει ομαλά με την λήψη των signals SIGINT, SIGQUIT.

2)ΛΕΙΤΟΥΡΓΙΑ CLIENT:

Αρχικά η main του προγράμματος ελέγχει οτι τα αρχικά arguments που απαιτούνται(και μπορεί να έρθουν με οποιδήποτε σειρά) είναι έγκυρα και αν ναι προχωράει.

Έπειτα ο client αναλαμβάνει αρχικά να δηλώσει την ύπαρξη του στον server στέλνωντας τα μηνήματα 1) LOG_ON <IP address, portNum> και 2) GET_CLIENTS.
Στην συνέχεια δημιουργεί την λίστα clientList καθώς και τον κοινόχρηστο κυκλικο buffer cbp για τοποθέτηση αντικειμένων tuple που τυχών λήφθηκαν απ'τον server.
Κατόπιν δημιουργεί worker threads τα οπόια αναλαμβάνουν να διαχειριστούν τα αντικείμενα που εισάγωνται στον κυκλικό buffer και τέλος αναλάμβανωντας τον ρόλο ενός P2P server μπαίνει σε έναν ατέρμονο βρόγχο όπου μπορει να εξυπηρετήσει 2 μηνύματα απ'τον server:
1) USER_ON <IP address, portNum> και 2) USER_OFF <IP address, portNum>.
Και 2 μηνύματα απο τα threads των υπόλοιπων client :
1) GET_FILE_LIST και 2) GET_FILE <pathname, version>
Τα worket threads ανάλογα με τον τύπο tuple που βρίσκουν στον κοινόχρηστο κυκλικό buffer στέλνουν τα εξής μηνύματα στην εφαρμογή client του remote peer :
1) GET_FILE_LIST και 2) GET_FILE <pathname, version>
αφού έχουν ελέξχει την τρέχουσα έκδοση του αρχείου που έχουν καλώντας μια hash function και τέλος για κάθε αρχείο που λαμβάνουν απτόν remote client δημιουργούν τα αντίστοιχα αντίγραφα στον τοπικό τους φάκελο και σε ξεχωριστό υποφάκελο για κάθε διαφορετικό client.
Τέλος ο client τερματίζει ομαλά με την λήψη των signals SIGINT, SIGQUIT και πριν κλείσει στέλνει στον server το μήνυμα LOG_OFF για να τον ενημερώσει, και αυτός με την σειρά του να ενημερώσει τους υπόλοιπους clients στο σύστημα.

COMPILING + EXECUTION:

Ανοίγουμε ένα terminal για compile
$ make
Ανοίγουμε 1 terminal για να σηκώσουμε τον server και τρέχουμε την παρκάτω εντολή
$ ./dropbox_server –p portNum
Ανοίγουμε 1 terminal για κάθε client που θέλουμε να προσθέσουμε στο σύστημα και τρέχουμε την παρακάτω εντολή για τον καθένα από τους n clients στο terminal window του.
$ ./dropbox_client –d dirNameN –p portNumN –w workerThreadsN –b bufferSizeN
serverPort –sip serverIP
(με οποιαδήποτε σειρά αρκεί να υπάρχουν όλα)
$ make clean (για clenaup στο τέλος)
Και για το script
1)$ ./create_files_small .sh dir_name num_of_files num_of_dirs levels

ΥΛΟΠΟΙΗΘΗΚΕ ΣΕ:

Language: C++(version 11), C, Unix(Bourne again)
OS: Ubuntu 18.10 (Cosmic Cuttlefish)
Text Editor: Sublime Text 3
Debugger: Valgrind 3.13.0, gdb