

2^η Εργασία

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΣΓΟΥΡΔΑΣ ΔΗΜΗΤΡΙΟΣ

ΑΜ: 1115201800173

Το πρόγραμμα δημιουργεί μία ιεραρχία διεργασιών ακριβώς όπως αυτή περιγράφεται στην εκφώνηση της εργασίας. Η συνολική εικόνα αποτελείται από έναν πατέρα και `numMonitors` παιδιά λαμβάνεται ως όρισμα.

Η κύριως λειτουργικότητα που χρειάζεται επεξήγηση είναι το πρωτόκολλο που έχει σχεδιαστεί, για την ομαλή επικοινωνία όλων των διεργασιών. Συγκεκριμένα, ο τρόπος με τον οποίο δουλεύει είναι πολύ *straight forward*. Το `buffer` αρχικοποιείται με χαρακτήρες `'.'` καθώς και θεωρείται κενή οποιαδήποτε θέση του `buffer` τον περιέχει. Πέρα από αυτό, κάθε δεδομένο τελειώνει/διαχωρίζεται με τον χαρακτήρα `'~'`, ώστε να μπορούμε να ξεχωρίσουμε τα διάφορα δεδομένα μεταξύ τους. Πέρα από αυτά, η άλλη ιδιαιτερότητα που έχει είναι ότι όταν πρόκειται να λάβουμε πλήθος δεδομένων απροσδιόριστου μεγέθους, η λήξη τους σηματοδοτείται με τον χαρακτήρα `'|'`. Για παράδειγμα όταν λαμβάνουμε τα `bloom filters`. Μπροστά από κάθε `bloom filter`, στέλνεται το όνομα του ιού στο οποίο αναφέρεται, έπειτα θα βρούμε τον διαχωριστικό χαρακτήρα `'~'` και αντίστοιχα έπειτα για το διάβασμα του `bloom filter`. Μόλις διαβάσουμε τον χαρακτήρα `'|'` γνωρίζουμε ότι έχουμε λάβει όλα τα `bloom filters` από την συγκεκριμένη διαδικασία, οπότε μπορούμε να προχωρήσουμε με τις υπόλοιπες εργασίες μας.

Για να μην διακόπτεται καμία κρίσιμη διαδικασία όταν λάβουμε ένα `signal` έχουμε αρχικά την `SA_RESTART` flag στην `sigaction` για να ολοκληρώνονται σωστά τα `syscalls`. Πέρα από αυτό, όταν λαμβάνουμε ένα `signal` απλά ανεβάζουμε ένα ανάλογο `flag` το οποίο κοιτάμε συνέχεια (για αυτό και η χρήση της `poll` στο `stdin`), όταν περιμένουμε για κάποια εντολή.

Έχει γίνει προσπάθεια στο να χρειαζόμαστε όσο το δυνατό λιγότερους `buffers` να διαβάσουμε για κάθε δουλειά, ώστε να αποφύγουμε τυχόν σπατάλες σε χώρο και χρόνο. Αυτό έχει επιτευχθεί με την χρήση των `custom` διαχωριστικών χαρακτήρων που περιγράφηκαν παραπάνω, ώστε να μπορούμε να αντιγράψουμε (και μετά να διαβάζουμε) κάθε πληροφορία `byte` προς `byte`, μέχρι να γεμίσει ο `buffer`, χωρίς κανένα πρόβλημα στην αναγνώριση της πληροφορίας. Έτσι επίσης, δεν χρειάζεται να στέλνονται άλλες `extra` πληροφορίες που θα έπαιρναν χώρο όπως το `size` ενός μηνύματος που ακολουθεί.

Για να μην υπάρχει χαμένος χρόνος σε κανένα στάδιο, έχει χρησιμοποιηθεί ένας συνδυασμός των `select` και `poll` συναρτήσεων (ανάλογα την περίπτωση), για να ελέγχουμε πότε ένα `buffer` είναι έτοιμο για διάβασμα.

Αξιοσημείωτο είναι το γεγονός ότι το πρώτο μήνυμα που θα σταλεί στα παιδιά έχει ένα προσυμφωνημένο `buffer size`. Αυτό το μήνυμα αποτελεί και το κανονικό `buffer size` το οποίο δίνεται από τα ορίσματα στον πατέρα. Αυτή η διαδικασία είναι απαραίτητη για να μάθουν τα παιδιά το `buffer size` καθώς στα ορίσματα των παιδιών έχει επιτραπεί να περνάμε μόνο το `path` των `pipes`.

Επίσης, όταν ένα παιδί γίνεται `resurrect` να σημειωθεί ότι δεν ξαναστέλνει τα `bloom filters` etc. καθώς δεν ζητείται από την εκφώνηση και δεν χρειάζεται, όπως επίσης και δεν υπάρχει λόγος για επιπλέον χρονοβόρες διαδικασίες.

Τέλος, το πρόγραμμα διαχειρίζεται όλη του την μνήμη δυναμικά, απελευθερώνοντάς την όποτε αυτή δεν χρειάζεται πλέον.

Τα προτότυπα των συναρτήσεων και οι βιβλιοθήκες βρίσκονται στο `functions.h`, ενώ το `structs.h` έχει τις δομές που χρησιμοποιούνται.

Για την ευκολότερη παραγωγή του των εκτελέσιμου, έχει δημιουργηθεί ένα `makefile` που χρησιμοποιεί `separate compilation`. Εκτελώντας την εντολή “`make`” στον κατάλογο της εργασίας παράγονται τα απαραίτητα εκτελέσιμα. Επίσης, για συντομία υπάρχει και το “`make run`” που τρέχει το πρόγραμμα με προκαθορισμένα ορίσματα. Τέλος, έχουμε και δυνατότητα για “`make clean`”, για καθαρισμό των εκτελέσιμων και `object files`.

Η υλοποίηση έχει γίνει σε C.

Όλες οι υπόλοιπες σημαντικές πληροφορίες βρίσκονται σε `comments` στον κώδικα.

Η συμπεριφορά του προγράμματος είναι ακριβώς αυτή που περιγράφεται στην εκφώνηση και συζητήθηκε στο Piazza. Η έξοδος είναι αυτή που φαίνεται από την εκφώνηση.

Το `bash script` είναι υλοποιημένο στο αρχείο `create_infiles.sh`, το οποίο βρίσκεται στον φάκελο `bash`. Καλείται με τη μορφή “`./create_infiles.sh data input_dir numFilesPerDirectory`” και παράγει το `output directory`, με όνομα ίδιο με το `input_dir`, όπως περιγράφεται από την εκφώνηση (προτεινόμενο όνομα για την εργασία είναι το “`dir`”, το οποίο θα πρέπει να μεταφερθεί στο ίδιο σημείο με τα `executables`). Η συμπεριφορά του είναι αυτή που ζητείται.