

2η Εργασία Λειτουργικά

Κωνσταντίνη Στόικου AM: 1115201500151

Το πρόγραμμά μου υλοποιεί όλα τα ζητούμενα της άσκησης.

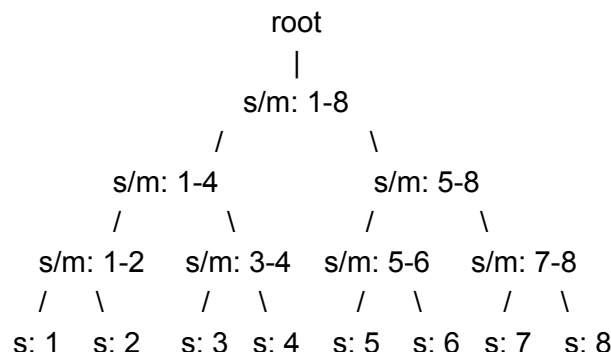
Τα αρχεία μου χωρίζονται ως εξής:

- root.c : το πρόγραμμα για την διεργασία του root
- root_functions.c, root_functions.h : συναρτήσεις για την root και το αντίστοιχο header
- splitter_merger.c : το πρόγραμμα για την διεργασία των ενδιάμεσων κόμβων
- spl_mer_functions.c, spl_mer_functions.h : συναρτήσεις για τον splitter/merger και το αντίστοιχο header
- searcher.c : το πρόγραμμα για την διεργασία των φύλων
- record.h : η δομή για τις εγγραφές
- statistic.h : η δομή για τον χρόνο, αποτελείται από έναν int που δείχνει αν αναφέρεται σε searcher ή splitter/merger και από μία μεταβλητή για τον χρόνο

Όταν δεν έχει δοθεί το skew σαν όρισμα τότε σε κάθε επίπεδο splitter/merger που δημιουργείται (εκτός του αρχικού που δημιουργεί η root) ο αριθμός των εγγραφών χωρίζεται στα δύο και ο ένας splitter/merger παίρνει (σαν παράμετρο μέσω exec) τις μισές εγγραφές και ο άλλος τις μισές αυξημένες κατά το mod της διαίρεσης.

Όταν έχει δοθεί το skew σαν όρισμα ο αριθμός των εγγραφών παραμένει ίδιος σε όλους τους κόμβους splitter/merger και αλλάζει (με βάση τον τύπο που δίνεται) όταν δημιουργούνται οι searchers (με την συνάρτηση breakNumOfRecords).

Συγκεκριμένα όταν δίνεται το skew σαν όρισμα η root περνάει στον πρώτο splitter/merger το range των id των searchers που θα δημιουργηθούν (id εννοώ αν είναι ο 1ος, 2ος κτλ searcher), π.χ. για ύψος 3 το range θα είναι 1-8 εφόσον θα δημιουργηθούν 8 φύλλα. Το range περνιέται με την μορφή 2 ξεχωριστών μεταβλητών start, end. Επίσης περνιέται σαν όρισμα το άθροισμα από 1 ως 2^h που χρησιμοποιείται στον τύπο του skew και παραμένει ίδιο σε όλους τους splitter/mergers. Σε κάθε splitter/merger χρησιμοποιείται η συνάρτηση calculateNewRange που σπάει κάθε φορά το εύρος των ids αναλόγως αν είναι ο πρώτος splitter/merger που γίνεται fork ή ο δεύτερος, π.χ. για ύψος 3 το εύρος ξεκινάει 1-8 και θα σπάσει σε 1-4 και 5-8, μετά σε 1-2, 3-4, 5-6, 7-8. Όταν το ύψος γίνει ίσο με 1 τότε ο ένας searcher παίρνει για id το start και ένας το end που θα τα χρησιμοποιήσουν για να σπάσουν τις εγγραφές ανάλογα με τον τύπο του skew. Π.χ. για ύψος 3



Το range ουσιαστικά δείχνει ποιοι searchers θα είναι εγγόνια του εκάστοτε splitter/merger. Ο τελευταίος searcher που δημιουργείται (στην προκειμένη περίπτωση, ο 8ος) θα διαβάσει επιπλέον και τις εγγραφές που περίσσεψαν λόγω mod από τους προηγούμενους searchers.

Οι splitter/merger μόλις διαβάσουν κάτι από το pipe που γράφει το παιδί το γράφουν κατευθείαν στον πατέρα. Ο κάθε searcher αφού γράψει τις εγγραφές γράφει μία τελευταία εγγραφή με custid -1 (επειδή οι κανονικές εγγραφές έχουν θετικά id) ώστε να ξέρει ο πατέρας ότι πρόκειται να

διαβάσει στατιστικό (χρόνο), και μετά γράφει το στατιστικό. Με την σειρά του ο πατέρας μόλις διαβάσει στατιστικό, γράφει στον πατέρα του μία εγγραφή με αρνητικό id και έπειτα γράφει το στατιστικό. Στο τέλος κάθε splitter/merger κάνει αντίστοιχη δουλειά για να γράψει το δικό του στατιστικό στον πατέρα του. Ο root διαβάζει τις εγγραφές και τους χρόνους που καταφθάνουν σε αυτόν. Τις εγγραφές τις γράφει σε ένα αρχείο results.txt που θα περαστεί σαν όρισμα όταν κληθεί η sort (και στο τέλος θα διαγραφεί). Τους χρόνους τους διαβάζει και αναλόγως αν προέρχονται από searcher ή splitter/merger βγάζει τους ανάλογους χρόνους min, max και average. Η root στο τέλος κάνει fork ώστε η διεργασία-παιδί της να καλέσει την sort που θα εκτυπώσει τα αποτελέσματα και αφού τελειώσει θα εκτυπωθούν τα στατιστικά, ο αριθμός των signals που κατέφθασαν στην root, ο συνολικός χρόνος για την ολοκλήρωση της ερώτησης και το πλήθος των εγγραφών που βρέθηκαν.

Χρησιμοποιώ απλά pipes γιατί θεώρησα ότι αυτά που προσφέρουν τα named pipes επιπλέον δεν μου χρησιμεύουν στην συγκεκριμένη άσκηση.

Πηγές που συμβουλευτήκα:

- <https://stackoverflow.com/questions/35683615/how-to-find-the-execution-time-of-code-in-c>
- <https://stackoverflow.com/questions/5248915/execution-time-of-c-program>
- <https://stackoverflow.com/questions/19461744/make-parent-wait-for-all-child-processes-to-finish>
- <https://stackoverflow.com/questions/18820006/writing-and-reading-multiple-times-on-same-pipe>
- <https://stackoverflow.com/questions/5189475/passing-a-pipe-as-a-parameter-in-exec>
- <https://stackoverflow.com/questions/29618839/how-to-send-the-pipe-from-parent-to-child-process>
- <https://stackoverflow.com/questions/7369286/c-passing-a-pipe-thru-execve>
- <https://stackoverflow.com/questions/2784500/how-to-send-a-simple-string-between-two-programs-using-pipes>
- <https://stackoverflow.com/questions/2708477/fork-and-wait-with-two-child-processes>
- <https://stackoverflow.com/questions/10669673/saving-off-the-current-read-position-so-i-can-look-for-it-later>
- <https://stackoverflow.com/questions/43419109/fork-tree-c-programs>
- <https://stackoverflow.com/questions/1695681/reading-from-and-writing-to-the-middle-of-a-binary-file-in-c>
- <https://stackoverflow.com/questions/2407605/c-warning-missing-sentinel-in-function-call>
- <https://www.geeksforgeeks.org/exec-family-of-functions-in-c/>