

ΙΩANNA ΛΥΚΟΥΔΗ
1115201400091

Τρόπος εκτέλεσης: ./jobExecutor -d paths.txt -w 2

Αρχεία:
jobExecutor.c
Arguments.c Arguments.h
Trie.c Trie.h

jobExecutor.c: Κατά την εκτέλεση του προγράμματος δίνεται ένα αρχείο με όλα τα paths που πρέπει να μοιραστούν στους workers. Ανοίγω το αρχείο και παίρνω μια γραμμή, δηλαδή κάθε path ξεχωριστά, και το αποθηκεύω στον πίνακα array. Έπειτα υπολογίζω πόσα ισάριθμα μονοπάτια θα πάρει ο καθένας, καθώς και πόσα περισσεύουν. Αφού δημιουργήσω τα κατάλληλα αρχεία για τα pipes, κάνω fork για να δημιουργήσω τα παιδιά. Η διαχείριση των διεργασιών μου έχει υλοποιηθεί σε ένα αρχείο. Ελέγχω το pid αν είναι παιδί ή αν είναι ο πατέρας.

Η υλοποίηση του παιδιού έχει ως εξής: Ανοίγει τα pipes για να μπορεί να επικοινωνήσει. Ξεκινάει και κάνει μια διαδικασία για όσο και ο αριθμός των μονοπατιών που θα του στείλει ο πατέρας. Έτσι ξέρει ότι τις πρώτες child_numdocs φορές αυτό που θα διαβάσει θα είναι μονοπάτι και το αποθηκεύει σε έναν πίνακα με τα μονοπάτια του παιδιού. Στην συνέχεια γράφει στον πατέρα "Next path" για να μπορεί να συγχρονιστεί και συνεχιστεί η διαδικασία. Στο σημείο λοιπόν που έχω λάβει το συγκεκριμένο μονοπάτι, βρίσκω όλα τα αρχεία που έχει μέσα το μονοπάτι. Κάθε τέτοιο αρχείο το ανοίγω, διαβάζω κάθε του λέξη και την εισάγω στο Trie αφού πρώτα υπολογίσω και πόσα bytes έχει. Επίσης, το παιδί γράφει κάποια στοιχεία από τις εντολές που δέχεται μέσω του πατέρα σε ξεχωριστό αρχείο στον φάκελο Log. Σε ένα άπειρο loop διαβάζει από τον πατέρα, αν το μήνυμα αντιστοιχεί σε κάποια εντολή τότε ενεργεί κατάλληλα. Αν πρόκειται για την /exit τα παιδιά τερματίζουν και τερματίζει και το πρόγραμμα. Αν πρόκειται για την /search τότε παίρνει μια μια τις λέξεις που δόθηκαν κατά την εκτέλεση της εντολής και τις αναζητά με την searchTrie. Τέλος, ελευθερώνει ότι έχει δεσμεύσει.

Η υλοποίηση του πατέρα έχει ως εξής: Μέσα στον έλεγχο για το pid της διεργασίας που δημιουργήθηκε ανοίγω τα pipes. Στην συνέχεια γίνεται το μοίρασμα των πακέτων. Πρώτα μοιράζω τα "ισάριθμα" μονοπάτια. Δίνω το πρώτο μονοπάτι στο πρώτο παιδί, μου απαντάει και έπειτα στέλνω το δεύτερο μονοπάτι στο δεύτερο παιδί κ.ο.κ. Στην συνέχεια μοιράζω ότι περίσσευε από τα μονοπάτια (αριθμός μικρότερος των παιδιών προφανώς). Παρακάτω, διαβάζει από το stdin τις εντολές μέσω του χρήστη. Αν πρόκειται για την /search τότε στέλνει την εντολή στο παιδί για να το ενημερώσει και στην συνέχεια στέλνει το string με τις λέξεις που πρέπει να ψάξει. Μόνο όταν όλα τα παιδιά απαντήσουν τελειώνει η διαδικασία. Αν πρόκειται για την /wc απλά αθροίζει τα δεδομένα που του στέλνει το κάθε παιδί και τα εκτυπώνει. Αν πρόκειται για την /exit τότε ενημερώνει πάλι τα παιδιά για ποια εντολή πρόκειται και αυτά στην συνέχεια τερματίζουν.

Φροντίζω στο τέλος του προγράμματος να διαγράψω τα αρχεία από τα pipes.

Trie.c: Η όλη υλοποίηση είναι ίδια με αυτή της προηγούμενης εργασίας. Το μόνο που προστέθηκε είναι στο struct του PostingList το μονοπάτι της εκάστοτε λέξης και διαμορφώθηκαν κατάλληλα οι συναρτήσεις.

Arguments.c: Η όλη υλοποίηση είναι ίδια με αυτή της προηγούμενης εργασίας. Πρόκειται για την συνάρτηση που “διαβάζουμε” τα ορίσματα από την γραμμή εντολών.