

Νίκος Κουτσοβασίλης 1115201500076

Κάποιες δομές όπως το trie και το map τα πήρα από την πρώτη μου εργασία (με μικρές αλλαγές όπου χρειαζόταν)

Στην main γίνεται το fork() των w workers και ανάλογα θα εκτελεστεί η jobExecutor() ή worker()  
Η jobExecutor στην αρχή μοιράζει τους καταλόγους του αρχείου που παίρνει σαν είσοδο κι έπειτα εκτελεί την interface() η οποία ζητάει από το χρήστη είσοδο και καλεί τις ανάλογες συναρτήσεις.

Η worker() αφού διαβάσει τους καταλόγους καλεί την Init() η οποία ανοίγει τα αρχεία που βρίσκει στους καταλόγους και φτιάχνει το map και το trie.

Έπειτα μπαίνει σε επανάληψη που διαβάζει το query από το pipe και στέλνει σε αυτό την απάντηση.

Τα pipes είναι Non-Blocking για διάβασμα από την jobExecutor. Το πρωτόκολλο είναι το εξής:

- ο worker περιμένει στο pipe (read)
- ο jobExecutor στέλνει σε όλους το query στο οποίο τα πρώτα bytes είναι ο αριθμός (int) που πρέπει να διαβαστούν, και έπειτα τα bytes του query
- ο worker διαβάζει πόσα bytes πρέπει να διαβάσει, ανοίγει το query το σπάει σε tokens και πράττει ανάλογα
- Αντίστοιχα μετά συνθέτει μία απάντηση γράφοντας πρώτα τον αριθμό των bytes .

Το deadline γίνεται κάνοντας pause και alarm ο jobExecutor για d δευτερόλεπτα. Μετά διαβάζει τα pipes, αν αυτά δεν περιέχουν κάτι τότε αυξάνει τον μετρητή των nonReply.

Κώδικας για την λήψης του SIGCHLD έχει γραφτεί ωστόσο δεν πρόλαβε να γίνει debug. Η λογική μου εκεί είναι να αλλάζω μέσα στον handler μία volatile μεταβλητή που μου λέει οτι κάποιος τερμάτισε, έπειτα όταν ελεγχω αυτή την μεταβλητή βρίσκω με waitpid ποιος τερμάτισε και κάνω ξανά fork και του στέλνω τα paths. Ο κώδικας είναι σχολιασμένος ωστόσο θεωρώ ότι αξίζει να βαθμολογηθεί έστω λίγο.

Για τα bash scripts πήρα βοήθεια όσον αφορά τον τρόπο σκέψης από έναν συμφοιτητή μου. Έχουν γίνει όλες οι λειτουργίες σε ένα script.