



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών



ΤΜΗΜΑ
ΠΛΗΡΟΦΟΡΙΚΗΣ &
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Μάθημα
Προγραμματισμός Συστήματος

Project 2

Όνομα
Κυλάφη Χριστίνα-Θεανώ

ΑΜ
1115201200077

Αθήνα, 2018

Η άσκηση έχει υλοποιηθεί σε γλώσσα C, στο πρόγραμμα “sublime text” ενώ παράλληλα δοκιμαζόταν σε περιβάλλον linux μέσω απομακρυσμένης σύνδεσης μέσω τερματικού macOS.

Στο φάκελο, περιλαμβάνονται τα εξής αρχεία εκτός από το readme :
makefile, server.c, structs.h, trie.c, trie.h, funs.c, funs.h , paths.txt (το αρχείο μονοπατιών με το οποίο έχει δοκιμαστεί επιτυχώς το εκτελέσιμο πρόγραμμα) , readmeProject2.

Με την εντολή “make” παράγεται το εκτελέσιμο “**jobExecutor**”. Η σύνταξη για την εκτέλεσή του, είναι:
./jobExecutor -d <όνομα αρχείου με τα μονοπάτια> -w <πλήθος workers>

Έχω προσθέσει ελέγχους λανθασμένης σύνταξης.

Το πρόγραμμα, ανοίγει το συγκεκριμένο αρχείο με τα μονοπάτια που έχει επιλέξει ο χρήστης, αποθηκεύει σε έναν πίνακα τα μονοπάτια (κάθε γραμμή κι ένα μονοπάτι), τα ισομοιράζει στους workers που έχει δημιουργήσει (αν το πλήθος workers είναι μεγαλύτερο από τα paths, δημιουργεί paths - πλήθος workers) και στη συνέχεια, ανάλογα με τις εντολές του χρήστη (/search, /mincount, /maxcount, ή /exit), στέλνει τα απαραίτητα δεδομένα στους workers, παίρνει απαντήσεις από αυτούς και παράγει τα αντίστοιχα αποτελέσματα, τυπώνοντας τέλος μήνυμα επιτυχημένου τερματισμού των παιδιών.

Το συγκεκριμένο πρόγραμμα βασίστηκε όσον αφορά τις δομές και τις συναρτήσεις στην 1η εργασία, γι’αυτό και περιλαμβάνω το readme και της πρώτης στον φακελο.

Παραδοχές & Ελλείψεις:

1. Η επιλογή του deadline(η /search εκτελείται κανονικά χωρίς ή και με την εισαγωγή του flag -d <deadline>) δεν έχει υλοποιηθεί. Θα αποθήκευα το timestamp στα παιδιά, στο σημείο που θα ολοκληρωνόταν η αναζήτηση του query, και θα το έστελνα μέσω fifo, στον πατέρα πριν την αποστολή των αποτελεσμάτων, ώστε αν είχε ξεπεράσει το επιθυμητό deadline που έδωσε ο χρήστης (timestamp παιδιου - timestamp of query arrival στον πατέρα > deadline) δε θα έπαιρνε τα αποτελέσματα από εκείνο το παιδι.
2. Έχω δημιουργήσει επιτυχημένα δομή που αποθηκεύει γραμμή γραμμή το κάθε αρχείο του κάθε φακέλου του κάθε path. Επίσης, κρατούσα λίστα με τις γραμμές στις οποίες εμφανίζεται μία λέξη, στο posting list της αντίστοιχης λέξης όμως κατά την υλοποίηση παρατήρησα μη αναμενόμενη συμπεριφορά της συγκεκριμένης δομής, οπότε και δε συνέχισα στην τελειοποίησή της(την έχω κρατήσει μέσα σε σχόλιο στον κώδικα). Εξ αιτίας αυτού όπως θα δείτε, στην /search, εμφανίζω μόνο τα paths στα οποία υπάρχει η λέξη και όχι τις γραμμές, εφόσον δεν είχα την υλοποίηση της λίστας αυτών των γραμμών. Έχω όμως αποθηκεύσει το πλήθος των γραμμών στις οποίες εμφανίζεται.
3. Το αρχείο loginfo.sh είναι ένα shell script το οποίο για να τρέξει, χρειάζεται να μεταφερθεί στον φάκελο “log” που δημιουργεί το πρόγραμμα ώστε να αποθηκεύσει τα logfiles που παράγουν οι workers. Το script αυτό, μετράει τις /search αναζητήσεις (keywords) που έχουν συνολικά βρεθεί σε όλους τους workers.
4. Δεν έχουν υλοποιηθεί τα σήματα. Εάν κάποιο παιδί τερμάτιζε, θα δημιουργούσα νέο, και θα του έδινα ακριβώς τα ίδια paths που είχε το προηγούμενο, εφόσον έχω αποθηκευμένα τα paths, και εύκολη τακτική διαμοιρασμού.
5. Ακολουθείται τακτική busy waiting (με blocking named pipes -fifos)
6. Ο κώδικας είναι εμπλουτισμένος με σχόλια για τα τεχνικά και λογικά θέματα και τακτικές.

Χωρισμός αρχείων:

Επέλεξα να χωρίσω τα αρχεία στις εξής κατηγορίες:

- trie: δημιουργία και καταστροφή trie και postings list.
- funs: οι λειτουργίες του προγράμματος df, tf, search, υπολογισμός score
- server: ο σκελετός του προγράμματος, ανάγνωση από stdin, οργάνωση λειτουργίας, κλπ
- structs: οι structs τύπου λίστας postings, trie.

Ενδεικτική εκτέλεση:

```
./jobExecutor -d paths.txt -w 3
```

```
/wc theatina
```

```
linux19:/home/users/sdi1200077/syspro/pro2>./jobExecutor -d paths.txt -w 3

Worker 0 (pid: 21015) started
Worker 1 (pid: 21016) started
Worker 2 (pid: 21017) started

Select mode:
1. "/search <query1> <query2> ... <query10> -d <deadline>"
2. "/maxcount <keyword>"
3. "/mincount <keyword>"
4. "/exit"
Enter command: /wc

-Word Count mode selected-

In Files:
Total Characters: 718
Total words: 163
Total Lines: 38
```

```
/search ti
```

```
Select mode:
1. "/search <query1> <query2> ... <query10> -d <deadline>"
2. "/maxcount <keyword>"
3. "/mincount <keyword>"
4. "/exit"
Enter command:
/search ti

-Search mode selected-

Keyword found in file: ./files/tempA/tempAA1/fileAA1_2
Keyword found in file: ./files/tempA/tempAA2/fileAA2_1
Keyword found in file: ./files/tempB/tempBB/fileBB1_1
Keyword found in file: ./files/tempC/FILEC1_3
```