Κ22: Λειτουργικά Συστήματα

Ονοματεπώνυμο: Κωνσταντίνος Πασχόπουλος

Αριθμός Μητρώου: 1115201500127

Η main της εργασίας μου βρίσκεται στο αρχείο myfind.c και οι τύποι που έχω ορίσει στο mytypes.h. Εκτός από το struct που αποθηκεύει τις εγγραφές με την μορφοποίηση της εκφώνησης, έχω χρησιμοποιήσει και δύο structs στα οποία βάζω χρόνους. Το struct times το χρησιμοποιούν οι leaf nodes για να στείλουν τον χρόνο που έκαναν να ψάξουν. Το struct timesSM το χρησιμοποιούν οι εσωτερικοί κόμβοι του δέντρου και έχει έξι μεταβλητές που συμβολίζουν τους min, max, και average χρόνους των φύλλων και τους αντίστοιχους χρόνους των splitter/mergers. Τους χρόνους που ζητούνται τους έχω μετρήσει με την συνάρτηση της c gettimeofday(), για να μετράω τον πραγματικό χρόνο που έχει περάσει και όχι cpu ticks ή user cycles.

root.c

Εδώ έχω υλοποιήσει την root node. Την root την καλώ με exec μέσω της myfind και δέχεται τα ορίσματα: όνομα αρχείου, το pattern που ψάχνουμε, το ύψος του δέντρο και αν θέλω ή δεν θέλω οι searchers να κάνουν ψάξουν ανομοιόμορφα τα αποτελέσματα.

Η επικοινωνία της με την ρίζα του δέντρου επιτυγχάνεται μέσω ενός named pipe. Μέσω της read διαβάζει όλα τα αποτελέσματα των searchers και τα γράφει σε ένα προσωρινό αρχείο, το output.txt, πριν στείλει αυτό το αρχείο στην sort. Η sort στη συνέχεια ταξινομεί τα αποτελέσματα και τα εκτυπώνει στο stdout.

Για να μετρήσω τα signals που θα στείλουν οι searchers δημιούργησα την συνάρτηση sig_handler, που διαχειρίζεται ένα σήμα όταν έρθει, και την μεταβλητή num_signals, η οποία μετράει πόσα σήματα καταφθάνουν συνολικά.

splitterMerger.c

Το εκτελέσιμο αυτό δημιουργεί το δυαδικό δέντρο αναδρομικά, μειώνοντας κάθε φορά το ύψος κατά 1. Δέχεται τα ορίσματα: όνομα αρχείου, pattern, το ύψος στο οποίο βρισκόμαστε κάθε φορά, skew, από πού ξεκινάει το αρχείο, πόσες εγγραφές έχει, πόσα φύλλα θα καταλήξει να έχει το δέντρο, ένα pipe για επικοινωνία με τον γονιό του κάθε φορά, το process id της ρίζας, την αρχή και το τέλος του range των φύλλων.

Αυτό το range το χρησιμοποιώ επειδή το ποιες εγγραφές θα ψάξει ο κάθε leaf node εξαρτάται από το ποιος είναι. Άρα για κάθε δυαδικό δέντρο αρχικά έχω το range: [1, 2^h]. Κάθε φορά που δημιουργώ κάποιο παιδί, αυτό θα παίρνει το μισό range από τον γονιό του. Έτσι ο κάθε leaf node καταλήγει με έναν μοναδικό αριθμό τον οποίο θα χρησιμοποιήσει για να βρει σε ποιες εγγραφές του αρχικού αρχείου θα πρέπει να ψάξει.

Όσο το ύψος δεν είναι 1 ο κάθε κόμβος δημιουργεί καινούργια παιδιά καλώντας τον εαυτό του με τα σωστά ορίσματα. Ο γονιός δέχεται, μέσω ενός named pipe που έχει με το κάθε παιδί, κάποια records και κάποιους χρόνους. Τα records και των δύο παιδιών τα στέλνει στο από πάνω επίπεδο. Τους χρόνους όμως πριν τους στείλει μπορεί να χρειαστεί να τους ανανεώσει με τον δικό του χρόνο.

Όταν το ύψος γίνει 1 καλεί το εκτελέσιμο των leaf nodes. Πριν το κάνει αυτό χρησιμοποιεί την συνάρτηση skew για να δώσει στο κάθε φύλλο ποιες εγγραφές να ψάξει.

leaf.c

Εδώ χρησιμοποιώντας μια for loop διαβάζει μια-μια τις εγγραφές στο διάστημα που έχει πάρει από το δέντρο και κάθε φορά που βρίσκει μια εγγραφή που ταιριάζει στέλνει ένα flag και μετά την γράφει στο pipe. Για την επικοινωνία μέσω των pipes έχω χρησιμοποιήσει 3 flags. Όταν πρόκειται να ακολουθήσει μία εγγραφή στέλνει τον αριθμό 1. Ένα θα ακολουθήσει ένα struct που περιέχει χρόνους στέλνει τον αριθμό 0. Στο τέλος στέλνει τον αριθμό -1 για να σημάνει ότι δεν έχει να στείλει άλλα δεδομένα μέσω του pipe.