

4η Εργασία Λειτουργικά

Κωνσταντίνα Στόικου AM: 1115201500151

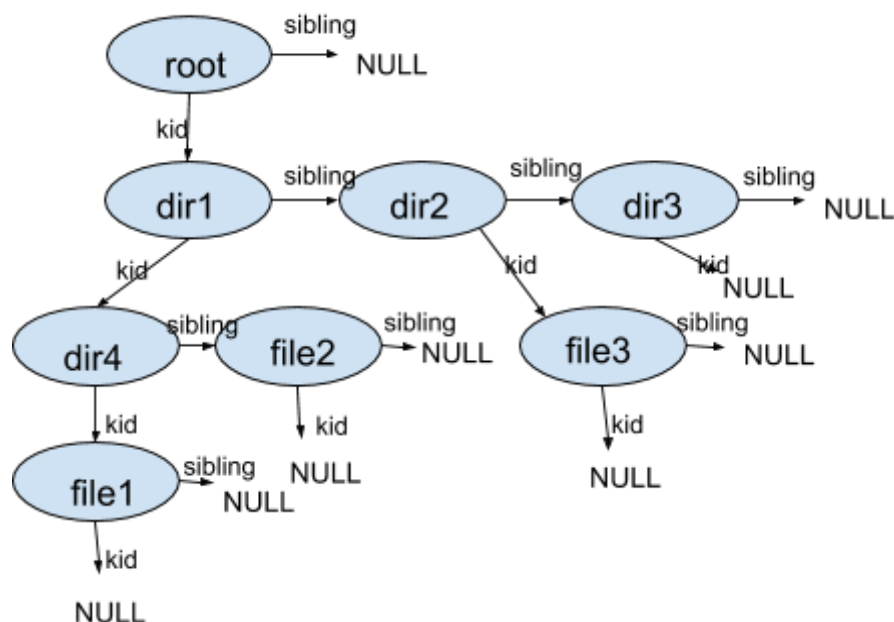
Αριστοφάνης Χιόνης Κουφάκος AM: 1115201500177

Κωνσταντίνα Στόικου :

Τα αρχεία που υλοποίησα εγώ είναι τα εξής:

- list.c : οι συναρτήσεις για την λίστα με τα i-nodes.
- namelist.c : οι συναρτήσεις για την λίστα με τα ονόματα των i-nodes.
- tree.c : οι συναρτήσεις για το δέντρο.
- updateTrees.c : οι συναρτήσεις για την ενημέρωση του δέντρου της πηγής με βάση τις αλλαγές που έγιναν στο file system.
- traverse.c : οι συναρτήσεις για τον αλγόριθμο συγχρονισμού των δέντρων.

Το δέντρο έχει υλοποιηθεί ως δυαδικό όπου ο ένας δείκτης δείχνει στην αρχή της λίστας των παιδιών του κόμβου (αν είναι φάκελος) και ο άλλος στον διπλανό του κόμβο που ανήκουν στον ίδιο φάκελο. Δηλαδή :



Οι φάκελοι dir1, dir2, dir3 βρίσκονται μέσα στον root.

Οι φάκελοι/αρχεία dir4, file2 βρίσκονται μέσα στον dir1.

Το αρχείο file1 βρίσκεται μέσα στον dir4.

Το αρχείο file3 βρίσκεται μέσα στον dir2.

Ο φάκελος dir3 είναι άδειος.

Για το list.c :

Η συνάρτηση addINode ελέγχει αν ο i-node του συγκεκριμένου path υπάρχει ήδη στην λίστα (που θα σημαίνει ότι το path είναι hardlink του υπάρχοντος κόμβου) και απλά προσθέτει το όνομα στην λίστα με τα ονόματα, αλλιώς δημιουργεί καινούργιο κόμβο.

Η συνάρτηση deleteINode κοιτάει αν στην λίστα με τα ονόματα του συγκεκριμένου κόμβου υπάρχει περισσότερα από ένα. Τότε απλά διαγράφει το όνομα από την λίστα, αλλιώς διαγράφει όλο τον κόμβο της λίστας.

Για το treeUpdates.c :

Η readDirectory “διαβάζει” τα αρχεία και τους φακέλους της πηγής στο file system και τα εισάγει μέσα στο δέντρο της πηγής. Οι υπόλοιπες συναρτήσεις καλούνται στην makeAction (eventHandlers.c) όπου ενημερώνουν το δέντρο της πηγής αναλόγως με το event που συνέβη (για

την μετακίνηση αρχείων ένα μέρος της ενημέρωσης του δέντρου γίνεται στην handleEvents στο eventHandlers.c).

Για το traverse.c :

Η συνάρτηση traverseTrees είναι μία αναδρομική συνάρτηση που έχει σαν βασικά ορίσματα δύο κόμβους, έναν από κάθε δέντρο. Σε κάθε αναδρομή δίνεται σαν όρισμα ο επόμενος κόμβος (κατά DFS) ή ο ίδιος αναλόγως αν προστέθηκε ή αφαιρέθηκε αρχείο. Για κάθε ζεύγος κόμβων που δίνεται σαν όρισμα καλείται η returnCase, που συγκρίνει τα ονόματα και τους τύπους των κόμβων (αν είναι αρχείο ή φάκελος) και επιστρέφει μία τιμή. Αναλόγως της τιμής, η traverseTrees θα κάνει ανάλογες ενέργειες για να συγχρονίσει τα δέντρα.

Αριστοφάνης Χιόνης Κουφάκος :

Τα αρχεία που υλοποίησα εγώ είναι τα εξής:

- eventActions.c : οι συναρτήσεις για κάθε action από το Action Table της inotify
- eventHandlers.c : οι συναρτήσεις για την αναγνώριση του event και κατάλληλη κλήση της ανάλογης συνάρτησης.
- functions.c : κάποιες συναρτήσεις γενικού σκοπού που χρησιμοποιούνται από διάφορα άλλα αρχεία.
- inotifyFunctions.c : οι συναρτήσεις για αρχικοποίηση και σωστό handling των watches της inotify.

Για το eventActions.c :

Έχω φτιάξει τις Mode συναρτήσεις εδώ, που υλοποιούν αυτά που λέει το table, κάνοντας τις αλλαγές στο φυσικό φάκελο του backup. Δηλαδή ανάλογα κάνουν create, delete, modify etc.

Για το eventHandlers.c :

Εδώ υπάρχουν πολλές βασικές συναρτήσεις όπως η handleEvents, η οποία είναι σα το παράδειγμα για την inotify που μας είχατε δώσει και ουσιαστικά είναι υπεύθυνη για το σωστό διάβασμα και μετάφραση των διαφόρων events της inotify. Αυτή τρέχει για πάντα μέχρι να δεχτεί Ctrl+C και να τερματίσει, και μόλις διαβάσει το event ολόκληρο, είναι έτοιμη να καλέσει την makeAction η οποία με τη σειρά της θα διαβάσει τη μάσκα του event και θα καλέσει την αντίστοιχη συνάρτηση Mode από το αρχείο με τα actions, ακόμα θα καλέσει τις σωστές συναρτήσεις για να διατρέξουμε τα δέντρα και να ενημερωθούν και οι λογικές δομές.

Για το functions.c :

Εδώ έχουμε συναρτήσεις γενικού σκοπού που χρησιμοποιούνται από πολλά αρχεία του προγράμματός μας. πχ η makeBackup είναι υπεύθυνη να φτιάξει το backup identical με το source , η formatBackupPath να φτιάξει το path για τα αρχεία του backup, δηλαδή να έχουν την ίδια διαδρομή που έχει το source, και το μόνο διαφορετικό να είναι το πρώτο string, αντί για source, backup.

Για το inotifyFunctions.c :

Εδώ έχουμε βασικές συναρτήσεις για αρχικοποιήσεις των watch descriptors. Για παράδειγμα, η recursiveWatch παίρνει ένα φάκελο και προσθέτει αναδρομικά σε όλους τους φακέλους μέσα της wd.