ΙΩΑΝΝΗΣ ΚΑΠΕΤΑΝΓΕΩΡΓΗΣ **Α.Μ. : 1115201800061**

K22: Λειτουργικά Συστήματα (Περιττοί) – Χειμερινό '20

Project 4

Γενικές παρατηρήσεις:

- Όλα τα αρχεία κώδικα είναι πλήρως σχολιασμένα και εξηγείται αναλυτικά η κάθε λειτουργία που γίνεται στο κάθε σημείο. Στο documentation αυτό, οι λειτουργιές που γίνονται σε κάθε σημείο εξηγούνται περιγραφικά. Οι προγραμματιστικές λεπτομέρειες εξηγούνται στα σχόλια του κώδικα έτσι ώστε αν γίνει πλήρως κατανοητός.
- Όση μνήμη έχει δεσμευθεί δυναμικά κατά την εκτέλεση του προγράμματος, αποδεσμεύεται πριν το τέλος αυτού με ελεγχόμενο τρόπο και χωρίς να υπάρχουν απώλειες μνήμης η errors (έχει ελεγχθεί μέσω του valgrind).
- Έχουν υλοποιηθεί όλα όσα ζητούνται στην εκφώνηση αλλά και αναφέρθηκαν στο piazza.
- Έχουν ελεγχθεί πολλές "ακραίες" περιπτώσεις έτσι ώστε το πρόγραμμα να έχει την επιθυμητή συμπεριφορά όποια και αν είναι η δομή των δεδομένων.
- Η διαχείριση των softlinks/hardlinks γίνεται κανονικά. Δηλαδή σε κάθε περίπτωση τα softlinks/hardlinks "κρατάνε" τις ιδιότητες τους στο destination folder.

Αρχεία που περιέχονται στο directory

- mainQuic.c : περιέχει την συνάρτηση main καθώς και την συνάρτηση traverseDir (η οποία καλείται από την main) και αποτελεί τον βασικό κορμό του προγράμματος quic καθώς είναι υπεύθυνη για το traversing στην ιεραρχία των directories.
- copy.c : το αρχείο αυτό αφορά την αντιγραφή αρχείων και softlinks. Η βασική συνάρτηση είναι η copyfile η οποία πραγματοποιεί την αντιγραφή των αρχείων (με την βοήθεια συναρτήσεων που επίσης περιέχονται στο αρχείο). Σε περίπτωση που θέλουμε να αντιγράψουμε ένα softlink, καλείται η συνάρτηση copySymLink η οποία αναλαμβάνει την αντιγραφή του.
- hardlinks.c: το αρχείο αυτό αφορά αποκλειστικά την αντιγραφή/δημιουργία των hardlinks. Η συνάρτηση που το υλοποιεί την λειτουργία αυτή είναι η checkHardLink. Για την διαχείριση των hardlinks χρησιμοποιώ μια linked list της οποίας η υλοποίηση υπάρχει επίσης στο αρχείο αυτό.
- delete.c: το αρχείο αυτό αφορά την διαγραφή αρχείων του destination directory όταν αυτά δεν υπάρχουν πλέον στο source directory. Δηλαδή, χρησιμοποιείται μόνο όταν δίνεται το flag -d κατά την εκτέλεση του προγράμματος. Η βασική συνάρτηση που υλοποιεί την λειτουργία αυτή είναι η checkForDeletedFiles. Η τελευταία χρησιμοποιεί την συνάρτηση deleteDirectory η οποία διαγράφει ένα directory (όταν αυτό χρειάζεται) και η υλοποίηση της επίσης υπάρχει στο αρχείο αυτό.
- include.h : το αρχείο αυτό περιέχει κάποια defines και κάποιες δηλώσεις τύπων/συναρτήσεων οι οποίες είναι κοινές για όλα τα παραπάνω αρχεία τα οποία κάνουν include το αρχείο αυτό.
- Makefile: εξηγείται αμέσως παρακάτω.

Makefile

- Εφαρμόζεται separate compilation. Αρχικά δημιουργούνται όλα τα .o αρχεία από τα αντίστοιχα .c αρχεία και έπειτα συνδέονται κατάλληλα έτσι ώστε να παραχθεί το αντίστοιχο εκτελέσιμο.
- Παράγεται το εκτελέσιμο πρόγραμμα με όνομα **quic**.
- Έχει προστεθεί και η λειτουργία make clean έτσι ώστε να διαγράφονται όλα τα object files και το εκτελέσιμο αρχείο.

Εκτέλεση προγράμματος

Η εκτέλεση του προγράμματος γίνεται ακριβώς όπως περιγράφεται στην εκφώνηση. Δηλαδή ως εξής:

./quic -v -d -l origindir destdir

Προφανώς μπορούμε να παραλείψουμε κάποια ή και όλες τις σημαίες (-v, -d, -l) αν δεν επιθυμούμε την αντίστοιχη λειτουργία.
Δεν έχουν προστεθεί επιπλέον σημαίες.

Σημαντική παρατήρηση:

Κατά την εκτέλεση του προγράμματος, προφανώς, χρησιμοποιώ κάποιους buffers στους οποίους κάθε φορά αποθηκεύω το τρέχων path των αρχείων/φακέλων.
Το μέγεθος του buffer αυτού γίνεται defined στο αρχείο include.h και έχει όνομα **MAX_PATH_SIZE**.
Το μέγεθος αυτό έχει γίνει defined σε **256**.
Σε περίπτωση που ο φάκελος που θέλουμε να αντιγράψουμε περιέχει μεγάλη ιεραρχία υποφακέλων, είτε σε περίπτωση που θέλουμε να δώσουμε σαν αρχικό path κατά την εκτέλεση ένα "μεγάλο" full path πρέπει να αυξηθεί το MAX_PATH_SIZE ανάλογα.
Γενικά, εάν υπάρχει περίπτωση σε κάποια φάση της εκτέλεσης το μήκος του path να υπερβεί το MAX_PATH_SIZE τότε πρέπει πριν την εκτέλεση το μέγεθος MAX_PATH_SIZE να αυξηθεί από το αρχείο include.h.

Περιγραφή λειτουργίας του quic

Τα “βήματα” που ακολουθούνται κατά την εκτέλεση του quic είναι τα εξής:

- Αρχικά, στην συνάρτηση main διαβάζονται οι σημαίες και τα paths τα οποία δόθηκαν κατά την εκτέλεση. Τα flags αποθηκεύονται σε global μεταβλητές έτσι ώστε όλες οι συναρτήσεις να γνωρίζουν αν πρέπει να εκτελέσουν τις αντίστοιχες λειτουργίες ή όχι. Τα paths (τα οποία μπορεί να είναι είτε relative είτε full) δίνονται σαν όρισμα στην συνάρτηση traverseDir η οποία και εκτελείται έτσι ώστε να “ξεκινήσει” η λειτουργία του quic.
- Η συνάρτηση traverseDir αρχικά ελέγχει εάν υπάρχει το destination directory το οποίο προσδιορίζεται από το path που δόθηκε σαν όρισμα. Εάν δεν υπάρχει το δημιουργεί. Αν υπάρχει τότε έχουμε τις εξής 2 περιπτώσεις:
  - Αν δεν έχει δοθεί το flag -d, τότε πρέπει να ελέγξουμε αν το αρχείο που αντιστοιχεί στο destination path είναι directory ή όχι. Αν δεν είναι directory, τότε διαγράφεται το αρχείο αυτό και δημιουργείται ένα directory με το ίδιο όνομα.
  - Αν έχει δοθεί το flag -d, τότε καλείται η συνάρτηση checkForDeletedFiles η οποία διαγράφει όσα αρχεία υπάρχουν στο destination directory και δεν υπάρχουν στο source directory. Η συνάρτηση checkForDeletedFiles εξηγείται αναλυτικότερα παρακάτω. Η λειτουργία που αναφέρθηκε αμέσως παραπάνω (δηλαδή στην περίπτωση που δεν έχουμε το flag -d) γίνεται εσωτερικά της checkForDeletedFiles.

  Στην συνέχεια, ελέγχεται αν με την αντιγραφή του Directory αυτού δημιουργείται κύκλος. Αν δημιουργείται, τότε δεν αντιγράφουμε αυτόν τον φάκελο.
  Αν δεν δημιουργείται κύκλος, τότε διατρέχουμε ένα προς ένα τα αρχεία που περιέχει το source directory.
  Για κάθε ένα αρχείο (ανεξαρτήτως τύπου) δημιουργούμε το αντίστοιχο path του. Δηλαδή για το αρχείο του source directory προσθέτουμε στο path του source directory το όνομα του αρχείου και αντίστοιχα για το αρχείο του destination directory.
  Στην συνέχεια καλούμε την συνάρτηση copyfile στην οποία δίνουμε σαν ορίσματα το path του αρχείου του source directory και το path του αρχείου του destination directory.
- Η συνάρτηση copyfile αρχικά εντοπίζει τον τύπο του source αρχείου. Ανάλογα με τον τύπο του αρχείου έχουμε τις εξής περιπτώσεις:
  - Αν το source αρχείο είναι ένα directory, τότε καλείται η συνάρτηση traverseDir για το source directory αυτό. Δηλαδή ακολουθείται η ίδια διαδικασία αναδρομικά (από το 2ο bullet) για τον υπό-φάκελο αυτό.
  - Αν το source αρχείο είναι ένα softlink, τότε σε περίπτωση που δεν έχει δοθεί το flag -l απλά αγνοείται. Αν έχει δοθεί το flag -l τότε καλείται η συνάρτηση copySymLink η οποία αναλαμβάνει την αντιγραφή του softlink. Η συνάρτηση softlink εξηγείται αναλυτικότερα παρακάτω.
  - Αν το source αρχείο είναι ένα hardlink, τότε σε περίπτωση που δεν έχει δοθεί το flag -l το αντιμετωπίζουμε σαν ένα απλό αρχείο (δηλαδή εκτελούμε όσα αναφέρονται στο αμέσως επόμενο βήμα). Αν έχει δοθεί το flag -l τότε καλείται η συνάρτηση checkHardLink η οποία αναλαμβάνει την διαχείριση των hardlinks. Η συνάρτηση checkHardLink εξηγείται αναλυτικότερα παρακάτω.
  - Αν το source αρχείο είναι ένα απλό αρχείο, τότε πρέπει να ελέγξουμε αν το αρχείο πρέπει να αντιγραφεί ή όχι. Αν το αρχείο που προσδιορίζεται από το destination path δεν υπάρχει, τότε προφανώς πρέπει να το δημιουργήσουμε και να αντιγράψουμε τα περιεχόμενα του source αρχείου. Η αντιγραφή γίνεται καλώντας την συνάρτηση deepCopy δίνοντας της σαν ορίσματα τα source file path και destination file path.
    Αν το αρχείο που προσδιορίζεται από το destination path υπάρχει τότε πρέπει να ελέγξουμε αν τα δύο αρχεία είναι ίδια. Έτσι, κάνουμε τους εξής ελέγχους:
    1. Αν τα δύο αρχεία είναι διαφορετικού τύπου
    2. Αν τα δύο αρχεία έχουν διαφορετικό μέγεθος
    3. Αν η ημερομηνία τροποποίησης του source file είναι νεότερη από αυτήν του destination file

    Σε περίπτωση που μία από αυτές της συνθήκες ισχύει, τότε τα δύο αρχεία δεν είναι ίδια και επομένως πρέπει να ανανεώσουμε το destination file. Ομοιώς με πριν καλούμε την συνάρτηση deepCopy έτσι ώστε να γίνει η αντιγραφή.
    Σε περίπτωση που καμία από τις τρεις συνθήκες δεν ισχύει τότε τα αρχεία είναι ίδια και έτσι δεν χρειάζεται να τα αντιγράψουμε.

Έτσι εφαρμόζοντας την παραπάνω διαδικασία για το source directory και αναδρομικά για όλα τα direcotries που περιέχονται σε αυτό, τελικά καταλήγουμε να έχουμε κάνει traverse όλη την ιεραρχία του source directory.
Πιο συγκεκριμένα, σε ένα “sum-up”, μέσω των αναδρομικών κλήσεων την traverseDir διατρέχουμε όλη την ιεραρχία του source directory και μέσω των κλήσεων των συναρτήσεων checkForDeletedFiles (όταν φυσικά υπάρχει το flag -d) και copyFiles (ή οποία με την σειρά στης καλεί την αντίστοιχες συναρτήσεις ανάλογα με τον τύπο του αρχείου) τελικά καταλήγουμε να έχουμε την αντίστοιχη επιθυμητή εικόνα στο destination directory.

checkForDeletedFiles:

Η συνάρτηση αυτή δέχεται σαν ορίσματα ένα source directory και ένα destination directory. Σκοπός της είναι να διαγράψει όσα αρχεία υπάρχουν στο destination directory αλλά όχι στο source directory.
Η συνάρτηση αυτή καλείται μόνο σε περίπτωση που από την γραμμή εντολών έχει δοθεί το flag -d κατά την εκτέλεση του quic. Η λειτουργία που εκτελείται είναι η εξής:
Ελέγχουμε κάθε ένα αρχείο του destination directory.
Για κάθε ένα αρχείο ελέγχουμε αν υπάρχει το αντίστοιχο αρχείο στο source directory. Αν το αρχείο δεν υπάρχει στο source directory τότε το αρχείο πρέπει να διαγραφεί. Ανάλογα με τον τύπο του αρχείου αυτού έχουμε τις εξής περιπτώσεις:

- Αν το αρχείο είναι directory, τότε καλείται η συνάρτηση deleteDirectory οι οποία είναι υπεύθυνη να διαγράψει τον φάκελο αυτό. Σκοπός της είναι να διαγραφούν πρώτα όλα τα αρχεία που περιέχονται στο directory αυτό έτσι ώστε να μπορούμε να διαγράψουμε και το ίδιο το directory. Η deleteDirectory διατρέχει ένα προς ένα τα αρχεία που υπάρχουν εσωτερικά του. Τα κανονικά αρχεία, τα softlinks και τα hardlinks απλά διαγράφονται. Αν υπάρχει ένας υπό-φάκελος εσωτερικά τότε η deleteDirectory καλείται αναδρομικά έτσι ώστε να διαγραφεί.
- Αν το αρχείο είναι ένα softlink και αν δεν έχει δοθεί το flag -l τότε απλά αγνοούμε το softlink και δεν το διαγράφουμε. Αν έχει δοθεί το flag -l τότε διαγράφουμε το softlink αυτό.
- Αν το αρχείο είναι κανονικό αρχείο ή hardlink τότε το διαγράφουμε με τον ανάλογο τρόπο. Τα hardlinks διαγράφονται και σε περίπτωση που δεν έχει δοθεί το flag -l καθώς στην περίπτωση αυτή τα hardlinks αναγνωρίζονται σαν απλά αρχεία (όπως θα εξηγηθεί και παρακάτω).

Η checkForDeletedFiles δεν ελέγχει ολόκληρη την ιεραρχία (δηλαδή και τους υποφακέλους) αλλά μόνο τα περιεχόμενα του φακέλου που δόθηκε σαν όρισμα. Τα περιεχόμενα των υποφακέλων (εάν αυτοί υπάρχουν) θα γίνει από επομένη κλήση checkForDeletedFiles που θα γίνει από την traverseDir που θα γίνει στην συνέχεια της εκτέλεσης του quic.

Διαχείριση softlinks:

- Αν δεν έχει δοθεί το flag -l

Στην περίπτωση αυτή τα softlinks αγνοούνται. Πιο συγκεκριμένα, αν υπάρχει ένα softlink στο source directory τότε αυτό δεν θα αντιγραφεί στο destination folder. Επίσης, αν υπάρχει ένα softlink στο destination directory το οποίο δεν υπάρχει στο source (π.χ. από κάποια προηγούμενη εκτέλεση του quic με το flag -l) τότε το softlink δεν θα διαγραφεί αν καλέσουμε τον quic με το flag -d, καθώς και πάλι το αγνοούμε.

- Αν έχει δοθεί το flag -l

Στην περίπτωση αυτή η αντιγραφή των softlinks γίνεται από την συνάρτηση copySymLink (όπως αναφέρθηκε και παραπάνω).Πιο συγκεκριμένα, η συνάρτηση αυτή αρχικά “διαβάζει” το softlink του source directory και κάνει τις αντίστοιχες αλλαγές στο path το οποίο “δείχνει” το softlink αυτό, έτσι ώστε να μετατραπεί στο path στο οποίο πρέπει να “δείχνει” το αντίστοιχο softlink του destination directory. (προφανώς ενδεχομένως να μην χρειάζεται κάποια αλλαγή στο path π.χ. αν το softlink “δείχνει” στο relative path: ./randomFile.txt)

Παρατήρηση: Έχουν ληφθεί υπόψιν όλες οι πιθανές περιπτώσεις έτσι ώστε πάντα η μετατροπή αυτή να γίνεται σωστά. Για παράδειγμα, μια “ακραία” περίπτωση είναι να έχουν δοθεί κατά την εκτέλεση τα full paths για τα source directory και destination directory και ένα softlink να “δείχνει” σε ένα relative path της μορφής: ../sourceDirectory/file.txt (το οποίο δεν έχει και πολύ λογική, αλλά παρόλα αυτά μπορεί να συμβεί). Ακόμα και σε αυτήν την περίπτωση η μετατροπή γίνεται σωστά έτσι ώστε το softlink του destination directory να δείχνει στο σωστό αρχείο. Επίσης έχουν ληφθεί υπόψιν όλοι οι πιθανοί συνδυασμοί όπου μπορεί κατά την εκτέλεση να έχουν δοθεί full/relative paths αλλά και τα links να περιέχουν full/relative paths. (περισσότερες λεπτομέρειες στα γενικά σχόλια της τελευταίας σελίδας)

Στην συνέχεια πρέπει να ελέγξουμε αν χρειάζεται να κάνουμε copy το softlink ή αν υπάρχει ήδη στο destination directory και είναι up to date (δηλαδή δεν χρειάζεται να το τροποποιήσουμε).
Έτσι ελέγχουμε αν υπάρχει το αντίστοιχο softlink στο destination directory. Αν δεν υπάρχει, τότε προφανώς δημιουργούμε ένα νέο το οποίο θα “δείχνει” στο path το οποίο “φτιάξαμε” παραπάνω.
Αν υπάρχει, τότε ελέγχουμε αν το path στο οποίο “δείχνει” είναι ίδιο με αυτό που “φτιάξαμε”. Αν είναι ίδιο, τότε το softlink δείχνει στο σωστό αρχείο και έτσι το αφήνουμε ως έχει. Αν δεν είναι ίδιο, τότε το “ανανεώνουμε” έτσι ώστε να δείχνει στο σωστό αρχείο.
Έτσι τελικά έχουμε το αντίστοιχο softlink στο destination directory το οποίο δείχνει στο σωστό αντίστοιχο αρχείο.

Τέλος, όσον αφορά την διαγραφή, σε περίπτωση που υπάρχει ένα softlink στο destination directory χωρίς να υπάρχει το αντίστοιχο softlink στο source directory, όταν κληθεί η checkForDeletedFiles τότε το softlink θα διαγραφεί κανονικά.

Παρατήρηση: Αν στο source directory υπάρχει ένα softlink το οποίο δείχνει σε ένα αρχείο εκτός του source directory, τότε το αντίστοιχο softlink δημιουργείται και στο destination directory δείχνοντας στο ίδιο path. Προφανώς, υπάρχει περίπτωση το softlink αυτό να είναι dangling. (αυτά προφανώς αν έχει δοθεί το flag -l)

Διαχείριση hardlinks:

- Αν δεν έχει δοθεί το flag -l

Στην περίπτωση αυτή τα χειριζόμαστε σαν απλά “αυτούσια” αρχεία. Δηλαδή ισχύουν όλα όσα έχουν ειπωθεί παραπάνω για τα κανονικά αρχεία. Ελέγχεται το αν πρέπει να αντιγραφούν από την συνάρτηση copyfile και σε περίπτωση που πρέπει να αντιγραφούν/ανανεωθούν αυτό γίνεται μέσω της συνάρτησης deepCopy.

Παρατήρηση: Σε περίπτωση που στο destination directory υπάρχουν ήδη κάποια hardlinks (π.χ. από προηγούμενη εκτέλεση του quic με το flag -l) και γίνει νέα κλήση του quic χωρίς το flag -l τότε τα hardlinks αυτά πρέπει να μετατραπούν σε “αυτούσια” αρχεία. Πιο συγκεκριμένα, η περίπτωση αυτή περιγράφηκε στο piazza από βοηθό του μαθήματος. Το πρόγραμμα μου έχει την συμπεριφορά που περιγράφηκε και απαιτείται στην περίπτωση αυτή αλλά και σε άλλες παρόμοιες περιπτώσεις. Παραθέτω το αντίστοιχο post του piazza το οποίο περιγράφει την συμπεριφορά την οποία έχω υιοθετήσει στο πρόγραμμα μου:

Αν έχουμε:
1. **./src/file (common file)**
2. **./src/file_hard (hardlink του παραπάνω file)**

κάνω quic -r -l σε ένα φάκελο dest οπότε έχω:
1. **./dest/file (common file)**
2. **./dest/file_hard (hardlink του παραπάνω file)**

μετά από λίγο ξανακάνω quic -r (χωρις -l) οπότε θα πρέπει in theory να έχω:
1. **./dest/file (common file)**
2. **./dest/file_hard (κανονικό αρχείο, όχι hardlink του παραπάνω file)**

και αν ξανατρέξουμε quic -r -l θα πρέπει να μας τα ξαναισιωσει τα links και να έχουμε:
1. **./dest/file (common file)**
2. **./dest/file_hard (hardlink του παραπάνω file)**

- Αν έχει δοθεί το flag -l

Στην περίπτωση αυτή αντιγραφή των hardlinks γίνεται από την συνάρτηση checkHardLink (όπως αναφέρθηκε και παραπάνω).Πιο συγκεκριμένα, η συνάρτηση αυτή χρησιμοποιεί μια linked list η υλοποίηση της οποίας βρίσκεται στο αρχείο hardlinks.c. Στην linked list κάθε ένας κόμβος περιέχει το path ενός αρχείου του source file, το αντίστοιχο inode του αρχείου αυτού καθώς και το path στο οποίο πρέπει να υπάρχει το αντίστοιχο αρχείο στο destination directory. Η διαδικασία που ακολουθείται είναι η εξής:

- Κάθε φορά που βρίσκουμε ένα αρχείο στην ιεραρχία του source directory το οποίο έχει παραπάνω από ένα links (δηλαδή το πεδίο st_nlink του inode είναι μεγαλύτερο του 1), καλείται η συνάρτηση checkHardLink για το αρχείο αυτό. Η κλήση της checkHardLink γίνεται ασχέτως αν το αρχείο είναι το “original” file (δηλαδή το αρχείο πάνω στο οποίο έχουν γίνει linked τα υπόλοιπα αρχεία) ή αν είναι ένα hardlink (δηλαδή έχει γίνει hardlinked στο original αρχείο).
- Για κάθε ένα αρχείο για το οποίο καλείται η checkHardLink αρχικά ελέγχουμε αν υπάρχει το αντίστοιχο αρχείο στο destination directory.
  - Αν δεν υπάρχει, για το αρχείο του source file που δόθηκε σαν όρισμα ελέγχουμε αν υπάρχει στην linked list κάποιο αρχείο του source directory με το ίδιο inode. Έτσι, έχουμε τις εξής δύο περιπτώσεις:
    - Αν δεν υπάρχει, τότε αρχικά προσθέτουμε το αρχείο αυτό στην linked list. Δηλαδή προσθέτουμε το path του αρχείου αυτού, το inode του καθώς και το path του αντίστοιχου αρχείου του destination directory. Επίσης αυτό σημαίνει ότι το αρχείο αυτό θα είναι το “original” file των αρχείων που είναι linked μεταξύ τους. Δηλαδή, το αρχείο γίνεται copied στο destination directory σαν κανονικό αρχείο (δηλαδή με νέο inode). Έτσι, η συνάρτηση επιστρέφει και η αντιγραφή ανατίθεται στην συνάρτηση copyfile.
    - Αν υπάρχει, τότε σημαίνει ότι στο source directory τα δύο αυτά αρχεία είναι hardlinked μεταξύ τους (αφού έχουν το ίδιο inode), καθώς και ότι στο destination directory υπάρχει το αντίστοιχο αντίγραφο του original file. Έτσι από την την linked list παίρνουμε το path του αντίστοιχου original αρχείου του destination folder και δημιουργούμε το νέο αρχείο το οποίο είναι ένα hardlink στο αρχείο αυτό.
      Δηλαδή έχουμε την εικόνα:
      1. **./src/file (common file)**
      2. **./src/file_hard (hardlink του παραπάνω file)**

      και
      1. **./dest/file (common file)**
      2. **./dest/file_hard (hardlink του παραπάνω file)**

- Αν το αντίστοιχο αρχείο υπάρχει στο destination directory, αρχικά πάλι ελέγχουμε για το αρχείο του source file που δόθηκε σαν όρισμα αν υπάρχει στην linked list κάποιο αρχείο του source directory με το ίδιο inode.
  - Αν δεν υπάρχει, σημαίνει ότι το αρχείο αυτό θα είναι το “original” file των αρχείων που είναι linked μεταξύ τους. Δηλαδή, πρέπει να ελέγξουμε αν το αρχείο του destination directory περιέχει τα ίδια δεδομένα με αυτό του source directory. Έτσι, παρόμοια με προηγουμένως αναθέτουμε την δουλειά αυτήν στην συνάρτηση copyfile καθώς τα “original” αρχεία τα χειριζόμαστε σαν απλά αρχεία. Επίσης, παρομοίως με πριν, προσθέτουμε το αρχείο αυτό μαζί με τις υπόλοιπες πληροφορίες στην linked list
  - Αν υπάρχει, τότε σημαίνει ότι στο source directory τα δύο αυτά αρχεία είναι hardlinked μεταξύ τους (αφού έχουν το ίδιο inode), καθώς και ότι στο destination directory υπάρχει το αντίστοιχο αντίγραφο του original file. Έτσι από την την linked list παίρνουμε το path του αντίστοιχου original αρχείου του destination folder και ελέγχουμε αν έχει το ίδιο inode με το αρχείο του destination directory για το οποίο κλήθηκε η συνάρτηση. Αν δεν έχουν το ίδιο inode σημαίνει ότι το original αρχείο του destination directory έχει τροποποιηθεί ή γενικότερα ότι πλέον τα δύο αρχεία του destination directory δεν είναι πια linked. Έτσι, διαγράφουμε το παλιό hardlink του destination directory και δημιουργούμε ένα νέο hardlink στο original αρχείο του destination directory.

Έτσι τελικά έχουμε διαχειριστεί κατάλληλα όλα τα πιθανά “σενάρια” όσον αφορά τα hardlinks και τελικά έχουμε πάντα την επιθυμητή εικόνα στο destination directory.

Γενικά σχόλια:

Έχουν ληφθεί υπόψιν οι περισσότερες (αν όχι όλες) οι ακραίες περιπτώσεις δεδομένων έτσι ώστε το πρόγραμμα να έχει την επιθυμητή συμπεριφορά σε κάθε μία από αυτές.
Κάποιες χαρακτηριστικές περιπτώσεις είναι:

- Έχουν ληφθεί υπόψιν όλα τα πιθανά paths τα οποία μπορεί να δοθούν από τον χρήση (π.χ. full paths ή relative paths) αλλά και περιπτώσεις όπου για παράδειγμα τα relative paths είναι της μορφής: ./folder ή ./folder/ ή folder/ ή folder.
- Κατά την αντιγραφή softlinks για μετατροπή του path από source σε destination έχει ληφθεί υπόψιν ότι το path του directory μπορεί να είναι relative η full άλλα και το path το οποίο περιέχει το softlink μπορεί να είναι επίσης relative η full. Επίσης, μπορεί ένα από τα δύο paths που έχουν δοθεί να είναι relative και το άλλο να είναι full path (π.χ. να έχει δοθεί full path για το source directory και relative για το destination directory). Γενικά έχουν ληφθεί υπόψιν όλοι οι πιθανοί συνδυασμοί (ακόμα και αυτοί που δεν έχουν λογική, αλλά παρόλα αυτά δεν απαγορεύεται να συμβούν) και σε κάθε περίπτωση το path μετατρέπεται σωστά έτσι ώστε το softlink του destination directory να δείχνει στο σωστό αρχείο του destination directory.
- Κατά την εκτέλεση πολλαπλών εκτελέσεων με εναλλάξ την ύπαρξη του flag -l, το πρόγραμμα έχει την επιθυμητή συμπεριφορά (όπως εξηγήθηκε και στην παράγραφο με των hardlinks).
- Αν εκτελεστεί αρχικά ο quic και έπειτα κάνουμε αλλαγές στο source directory, έχουν ληφθεί υπόψιν όλες οι “αλλαγές παγίδες” που μπορεί να γίνουν στο source directory έτσι ώστε μετά από την επόμενη εκτέλεση του quic να έχουμε την ίδια εικόνα και στο destination folder. Για παράδειγμα μια αλλαγή παγίδα είναι:

  Αν έχουμε:

  1. **./src/file (common file)**

  εκτελώ τον quic σε ένα φάκελο dest οπότε έχω:

  1. **./dest/file (common file)**

  αλλάζω το src directory σε:

  1. **./src/file/newFile (δηλαδή μετατρέπω το αρχείο σε directory με το ίδιο όνομα)**

  μετά ξαναεκτελώ quic οπότε πρέπει να έχω:

  1. **./dest/file/newFile**
- Κατά την εκτέλεση γίνονται οι κατάλληλοι έλεγχοι έτσι ώστε να μην δημιουργηθεί ποτέ ένας κύκλος. Για παράδειγμα η εκτέλεση `quic .. .` δημιουργεί έναν κύκλο. Γενικότερα, εάν το destination directory βρίσκεται εσωτερικά του source directory τότε δημιουργείται ένας κύκλος. Σε κάθε περίπτωση, η υλοποίηση μου το εντοπίζει και ο κύκλος αποφεύγεται κατάλληλα (όπως εξηγήθηκε και παραπάνω).

Εκτύπωση στατιστικών:

- Έπειτα από κάθε εκτέλεση εκτυπώνονται όλα τα στατιστικά που ζητούνται στην εκφώνηση
- Στον αριθμό των συνολικών αρχείων, περιέχονται και όλα τα directories (μαζί και το αρχικό directory το οποίο δόθηκε σαν όρισμα).
- Ομοίως στο αριθμό των copied entities περιέχεται και ο αριθμός των directories που δημιουργήθηκαν κατά την εκτέλεση του quic.
- Έχει προστεθεί και ο αριθμός των entities τα οποία διαγράφηκαν από το destination directory (όπως προτάθηκε στο piazza). Στο αριθμό αυτό περιέχονται και οι περιπτώσεις όπου στο destination directory υπάρχει ένα αρχείο με ίδιο όνομα αλλά διαφορετικού τύπου από το αντίστοιχό αρχείο του source directory και έτσι πρέπει να διαγραφεί πρώτου κάνουμε την αντιγραφή του νέου αρχείου. Η περίπτωση αυτή εξηγήθηκε αμέσως παραπάνω στα Γενικά σχόλια στο 4ο bullet.
- Ο αριθμός των deleted entities εκτυπώνεται μόνο όταν υπάρχει το flag -d.