

OS Project 4

Βησσαρίων Μουτάφης
1115201800119

SDI1800119@DI.UOA.GR

Notes για τους διορθωτές:

- Γίνεται χρήση της *lstat* **μόνο**.
- Μην διαγράψετε τα directories μέσα στο φάκελο της εργασίας, χρειάζονται για τα test-runs (δες στο επόμενο section).
- Αν έχουμε το path src/phony και dest/phony και το ένα είναι dir/file και το δεύτερο file/dir, διαγράφουμε το τελευταίο και κάνουμε κανονικά copy το πρώτο.
- Τα sym links δεν λαμβάνονται υπ όψιν όταν δεν έχω το flag "-l".
- Αν το path src/hardlink είναι ένα hardlink σε ένα path και τρέξουμε quic **χωρίς** το "-l", τότε το αντιμετωπίζουμε σαν απλό αρχείο. Επίσης, εφόσον υπάρχει ήδη, την επόμενη φορά που θα τρέξουμε quic και **θα βάλουμε το "-l"**, **θα μετατραπεί σε hard link**.
- Αν το dest dir **υπάρχει ήδη**, τότε δεν το μετράμε στα copied items οπότε αν έχω λόγου χάρη 18 στοιχεία στο src hierarchy μαζί με το src dir, τότε τα statistics θα δείξουν 17/18 copied/detected items.
- Στα στατιστικά τα total items in hierarchy είναι πάντα αυτά του src. Λαμβάνω υπ όψιν και τα directories.

Compilation and Run

Παρέχω ένα Makefile που φροντίζει για το separate compilation των modules που χρησιμοποιεί το πρόγραμμα. Για να τρέξετε το πρόγραμμα τρέξτε τις παρακάτω εντολές.

```

1 ~$ make # compile the program and copy an instance of the executable at root
  directory.
2 ~$ make run # compile the program and run a test input (do not delete any of
  the directories in the root dir)
3 ~$ make clean # uninstall the program
4 ~$ make clean--dest # clean the dest dir of the test run

```

Listing 1: Compilation and Run Commands

Abstract

Η εφαρμογή *quic*, έχει υλοποιηθεί έτσι ώστε να αντιγράφει αναδρομικά ένα *directory* από το δοσμένο *path*. Διαθέτει 3 flags τα οποία ελέγχουν το *verbosity*, την διαγραφή των φακέλων που δεν υπάρχουν στο *src dir* και την διαχείριση των *links* (*soft & hard*).

Συγκεκριμένα αν το *dest-dir* είναι άδαιο και το "-l" **δεν έχει τεθεί** τότε πολύ απλά ακολουθούμε όλα τα *hard links* και φτιάχνουμε τους φακέλους. Αν όμως έχουμε όλα τα *links* έτοιμα (από προηγούμενο *run*) τότε δεν πειράζουμε τιποτα καθώς το τελικό αποτέλεσμα θα είναι ίδιο στον *end user*. Επίσης, σε κάθε περίπτωση, **αν θέσουμε** το "-l", τότε, *αν το αντίστοιχο link δεν υπάρχει*, το δημιουργούμε, είτε υπάρχει ο αντίστοιχος φάκελος είτε όχι, οπότε στα *hard links* θα έχω διαγραφή του ξεχωριστού *i-node* και *point* στο *proper inode*. Τέλος, είναι υπεύθυνος ο χρήστης αν θα εκτελέσει σωστά το πρόγραμμα και δεν θα καταλήξει με *dangling symlinks* που να δείχνουν σε φακέλους εκτός του *src dir hierarchy*.

Utilities

Τα *utilities* αποτελούνται από μικρές συναρτήσεις, οι οποίες βοηθούν στον σωστό έλεγχο και το *manipulation φακέλων* (*copy, creation, delete, etc*) καθώς και στην ανίχνευση διάφορων *edge cases* (πχ *detect cycle in the hierarchy, check for deleted files/directories in the src directory, if needed, etc*). Επίσης μέσα σε αυτή έχει τοποθετηθεί και μία συνάρτηση για τον έλεγχο της ομοιότητας 2 αρχείων, καθώς και μικρές συναρτήσεις που δημιουργούν τα *links*. Συγκεκριμένα, υπάρχει μια απλή συνάρτηση που δημιουργεί *symlinks* και μία η οποία δημιουργεί *hard links*.

Για τα **hard links** διαθέτουμε ένα *map* με *entries < src st_ino, path to the repsective dest inode >*, οπότε την πρώτη φορά που συναντούμε ένα *src entry* με *inode number = n*, τότε θα το **εισάγουμε στο map**. Την επόμενη φορά απλά θα μπορούμε να τσεκάρουμε αν το *ino* είναι στο *map* και αφού το βρούμε, θα μπορούμε να το ενώσουμε με ένα σχετικό *path* στο *dest dir*.

Παράλληλα έχουμε υλοποιήσει και την δικιά μας έκδοση της εντολής *rm -rf* του *unix* καθώς **διαγράφουμε αναδρομικά τα directories** αναγνωρίζοντας αν το *path* που δίνεται είναι *directory/file* και δρώντας ανάλογα σε κάθε βήμα. Για τον έλεγχο των διαγεγραμμένων από το *src dir*, αρχείων ή και *directories* κάνουμε χρήση της *check_deleted*, την οποία τρέχουμε σε κάθε *level* στο *dest hierarchy* όπως κάνουμε το *copy* για να είναι αποδοτικός ο έλεγχος και να μην κάνουμε περιττά *system calls* σε όλο το *dest-hierarchy*.

File Manipulation Routines

Για το *file manipulation* έχουμε μια γενική συνάρτηση που δίνεις ένα *in* και ένα *out path* και κάνει το *copy*. Η ρουτίνα αυτή ελέγχει αν τα 2 *elements* διαφέρουν ως προς τους τύπους (δες και στα *notes 3o bullet*) και αναγνωρίζει αν το *src path* είναι *file*, οπότε το αντιγράφει (διαφορετική συμπεριφορά μέσα στην *copy file* αν έχω το *link extension*), ή αν είναι *directory*, όπου το αντιγράφει αναδρομικά.

Η συνάρτηση για αντιγραφή φακέλων υλοποιεί και το *link extension* αφού αν το flag *manage_links* είναι *true*, θα εντοπίσει και αντιγράψει τα *hard/sym links*. Για τα *hard links*, συγκεκριμένα παρέχω και ένα *map* ώστε να μπορώ να έχω επίγνωση των φακέλων που αντιγράφηκαν κάθε

στιγμή εκτέλεσης, ώστε να τα ενώνω σωστά και αν ψάχνω αποδοτικά για την πληροφορία που χρειάζομαι (υπάρχει το inode ώστε να το ενώσω ή πρέπει να το φτιάξω;).

Για directory copy παρέχουμε μια αναδρομική συνάρτηση όπου ουσιαστικά κάνει χρήση της γενικής copy routine για να παρέχει generic copy behaviour. Επίσης μέσα σε αυτή γίνεται και ο έλεγχος για διαγεγραμμένα από το src, στοιχεία τα οποία φροντίζει αν διαγράψει με τον τρόπο που πρέπει (αντίστοιχα για directories και για files/links).

Hash Table with Double Probing

Το hash table που παρέχεται για την διαχείριση των hard links, είναι υλοποιημένο με ως ένα array από void pointers. Ο χρήστης παρέχει key-item creation, deletion, comparison και hashing functions έτσι ώστε να μπορεί να λειτουργήσει και να απελευθερώσει την μνήμη σωστά. Το double probing γίνεται με την χρήση της συνάρτησης *probing_hash(key, ht_size)*, η οποία κατανέμει ομοιόμορφα τα probes ώστε να πετύχουμε καλύτερη κατανομή στο hash table. Παρέχονται όλες οι βασικές λειτουργίες *create, insert, delete, contains, destroy*. Επίσης παρέχεται και μια συνάρτηση για την εκτύπωση των κλειδιών (debugging purposes).

References

- Διαφάνειες του μαθήματος.
- Διάφορα posts του stack overflow που εξηγούσαν γιατί πρέπει να χρησιμοποιήσω lstat για να παίζω με links.
- Manpages για τις διάφορες system calls για την κατανόηση των error codes που επιστρέφουν και για να καταλάβω πως λειτουργούν εξ ολοκλήρου.