

Λειτουργικά Συστήματα
Εργασία 4
Νίκος Ευτυχίου
sdi1800221

- Γενικές πληροφορίες σχετικά με τον πρόγραμμα:

Το πρόγραμμα επιλέχθηκε να δομηθεί σε 7 αρχεία για την ευκολότερη κατανόηση και σαφήνεια του κώδικα.

Για να γίνει compile το πρόγραμμα χρησιμοποιήστε την εντολή make.

Το πρόγραμμα quic πρέπει να δεχθεί σαν είσοδο από την γραμμή εντολών την παράμετρο "-f source dest" όπου source και dest ο φάκελος για αντιγραφή και το μονοπάτι που θα γίνει η αντιγραφή αντίστοιχα. Επίσης στο πρόγραμμα μπορούν να δοθούν οι παραμέτροι "-d" , "-l" , "-v" όπου διαγράφει περιττά αρχεία που υπάρχουν στο dest αλλά όχι στο source, λαμβάνει υπόψη τα links που ενδέχεται να υπάρχουν και εκτυπώνει κάποιες επιπλέον πληροφορίες σχετικά με το τι συμβαίνει στο πρόγραμμα.

Αρχείο quic.c:

Η main συνάρτηση αρχικά καλεί την συνάρτηση parametersCheck για να ελέγξει αν έχουν δοθεί σωστά τα ορίσματα στο πρόγραμμα. Στη συνέχεια μέσω της getParametersValues παίρνει τα ορίσματα που έχουν δοθεί στο πρόγραμμα, με την βοήθεια της opendir ελέγχει αν το μονοπάτι που έχει δοθεί για αντιγραφή υπάρχει αλλιώς το δημιουργεί με την CheckDest. Έπειτα αν υπάρχει η "-d" τότε καλείται η CheckDeletedFiles για να σβήσει όποια αρχεία ή και φακέλους δεν χρειάζονται. Μετά με την CopyFile ξεκινά η κύρια λειτουργία του προγράμματος, δηλαδή η αντιγραφή των αρχείων. Τέλος εκτυπώνονται πληροφορίες σχετικές με την quic.

Αρχείο delete.c:

Συνάρτηση void CheckDeletedFiles(char *origin, char *dest, int verbose)

Η συνάρτηση αρχικά ανοίγει το directory destination και σε ένα βρόγχο διαβάζει όλα τα αρχεία που βρίσκονται στο destination. Σε ένα δεύτερο βρόγχο ανοίγουμε το directory source για να διαβάσουμε επίσης όλα τα αρχεία του και να τα συγκρίνουμε με τα αρχεία στο destination. Αν δύο αρχεία έχουν ίδιο όνομα τότε γίνεται έλεγχος αν είναι directories και τα δύο, όπου δημιουργείται το νέο μονοπάτι για source και destination για να κλειθεί αναδρομικά η συνάρτηση και τα ελέγξει το subdirectory. Αν τα δύο αρχεία έχουν διαφορετικό τύπο ή αν ένα αρχείο υπάρχει μόνο στο destination directory τότε διαγράφεται. Αν ο τύπος είναι directory για διαγραφή τότε η RemoveAll φρονίζει να έχουν διαγραφεί όλα όσα βρίσκονται μέσα στο directory.

Συνάρτηση void RemoveAll(char *path, int verbose)

Η συνάρτηση με τις συναρτήσεις opendir και readdir παίρνει όλα τα αρχεία και τα διαγράφει από το directory. Αν υπάρχει κάποιο subdirectory τότε διαγράφεται αναδρομικά με την RemoveAll ξανά.

Αρχείο copy.c:

Συνάρτηση int CopyFiles(char *origin, char *dest, char *initialOrigin, char *initialDest, int l, long int *bytes, int verbose)

Η συνάρτηση αρχικά ανοίγει το directory source και σε ένα βρόγχο διαβάζει όλα τα αρχεία που βρίσκονται στο source. Σε ένα δεύτερο βρόγχο ανοίγουμε το directory destination για να διαβάσουμε επίσης όλα τα αρχεία του και να τα συγκρίνουμε με τα αρχεία στο source.

Αν δύο αρχεία έχουν ίδιο όνομα τότε γίνεται έλεγχος αν είναι directories και τα δύο, όπου δημιουργείται το νέο μονοπάτι για source και destination για να κλειθεί αναδρομικά η συνάρτηση και τα ελέγξει το subdirectory.

Αν δύο ομόσημα αρχεία έχουν ίδιο τύπο τότε δημιουργούμε το μονοπάτι για τα δύο αρχεία και ελέγχουμε αν έχουν ίδιο μέγεθος και ίδια ημερομηνία τελευταίας τροποποίησης. Αν κάποιο από τα δύο δεν συνάδει τότε διαγράφεται το αρχείο από το destination και καλούμε την Copy για να δημιουργήσει το νέο αρχείο.

Αν δύο αρχεία δεν έχουν ίδιο τύπο και το αρχείο στο destination είναι directory τότε δημιουργούμε το μονοπάτι για τα δύο αρχεία, με την RemoveAll διαγράφουμε το directory destination και με την Copy αντιγράφουμε το αρχείο.

Αν δύο αρχεία δεν έχουν ίδιο τύπο και το αρχείο στο source είναι directory τότε δημιουργούμε το μονοπάτι για τα δύο αρχεία, διαγράφουμε το destination, δημιουργούμε το directory destination. και καλούμε αναδρομικά την CopyFiles.

Αν δύο αρχεία δεν έχουν ίδιο τύπο και δεν είναι directories τότε δημιουργούμε το μονοπάτι, διαγράφουμε το destination και με την Copy αντιγράφουμε το αρχείο.

Τέλος αν το αρχείο δεν υπάρχει στο Destination τότε δημιουργούμε το νέο μονοπάτι, ελέγχουμε αν είναι directory και αν είναι δημιουργούμε το νέο directory και καλούμε την CopyFiles για να αντιγράψει τα αρχεία. Σε αντίθετη περίπτωση με την Copy αντιγράφουμε το αρχείο στο destination και επιστρέφει τον αριθμό των αρχείων που έχουν ελεγχθεί .

Συνάρτηση Copy(char *source, char *dest, char *initialSource, char *initialDest, int l, int verbose))

Αρχικά η συνάρτηση ελέγχει αν το source όπου έχει δοθεί είναι symbolic link και αν έχει δοθεί η "-l". Αν ναι, τότε συγκρίνει το absolute path του source directory και το absolute path στο οποίο 'δείχνει' το symbolic link. Αν το δεύτερο είναι substring του πρώτου τότε το link βρίσκεται μέσα στο source και δημιουργούμε το κατάλληλο μονοπάτι για το destination. Αν δεν βρίσκεται μέσα στο source τότε το link είναι ίδιο. Στη συνέχεια δημιουργεί το symbolic link και με την lutimes εξισώνει το last accessed date και last modified date των δύο αυτών links.

Αν στο αρχείο υπάρχουν περισσότερα από ένα hardlinks με τη βοήθεια της CheckHardlink ελέγχουμε αν όλα τα hardlinks σε αυτό το inode βρίσκονται μέσα στο source. Αν όχι τότε κάνουμε απλά link το αρχείο. Σε αντίθετη περίπτωση ψάχνουμε αν υπάρχει το link στο destination. Αν υπάρχει τότε κάνουμε link το νέο αρχείο αλλιώς κάνουμε deep copy στο destination.

Σε οποιαδήποτε άλλη περίπτωση κάνουμε deep copy το αρχείο με την cp.

Τέλος με την utimes εξισώνει το last accessed date και last modified date των δύο αυτών αρχείων.

Συνάρτηση int cp(char *source, char *dest))

Η συνάρτηση ανοίγει το source αρχείο για να το διαβάσει και το destination αρχείο για να γράψει σε αυτό ό,τι έχει διαβάσει από το source. (Η συνάρτηση έχει ίδια βάση με την συνάρτηση του x. Α. Δελη στις διαφάνειες και έχει τροποποιηθεί κατάλληλα για τους σκοπούς αυτής της εργασίας).

Αρχείο misc.c:

Συνάρτηση int parametersCheck(int argc, char * argv[])

Η συνάρτηση ελέγχει αν τα ορίσματα που έχουν δοθεί στο πρόγραμμα είναι ορθά.

Συνάρτηση void getParametersValues(int *v, int *d, int *l, char **origin, char **dest, int argc, char *argv[])

Η συνάρτηση ελέγχει και παίρνει τα ορίσματα που έχουν δοθεί και τα τοποθετεί σε κατάλληλες μεταβλητές.

Συνάρτηση void CheckDest(char *Destination)

Η συνάρτηση παίρνει σαν όρισμα ένα string. Με την opendir ελέγχει αν αυτό το είναι valid path. Αν δεν είναι τότε με την strtok συνάρτηση παίρνει ένα προς ένα τα κομμάτια του Destination και τροποιώντας τα κατάλληλα ελέγχει αν αυτά είναι valid paths αν δεν είναι τότε δημιουργεί το path.

Συνάρτηση int CheckHardlink(char *root, int inode, char **pathToFile)

Η συνάρτηση με την opendir και readdir ελέγχει αν κάποιο αρχείο στο φάκελο root έχει ίδιο inode με το όρισμα inode. Αν ναι τότε ο συνολικός αριθμός των αρχείων με ίδιο inode αυξάνεται και by reference εισάγουμε αυτό το μονοπάτι στο pathToFile. Αν το αρχείο είναι directory τότε καλούμε την συνάρτηση αναδρομικά. Τέλος η συνάρτηση επιστρέφει τον συνολικό αριθμό των αρχείων με ίδιο inode.

Συνάρτηση char *ReturnInodePath(char *initialSource, char *initialDest, int inode)

Η συνάρτηση με την opendir και readdir ελέγχει αν κάποιο αρχείο στο φάκελο initialSource έχει ίδιο inode με το όρισμα inode. Για κάθε αρχείο που ελέγχουμε δημιουργείται το κατάλληλο μονοπάτι. Σε περίπτωση όπου υπάρχει κάποιο subdirectory η συνάρτηση καλείται αναδρομικά. Η συνάρτηση επιστρέφει το μονοπάτι στο αρχείο του destination όπου είναι ίδιο με το αρχείο στο source και σε περίπτωση όπου δεν βρει τότε επιστρέφει NULL.