

**Ονοματεπώνυμο:** Θεοφανόπουλος Μιχαήλ

**Αριθμός Μητρώου:** 1115201800053

#### **4η Προγραμματιστική Άσκηση Λειτουργικά Συστήματα**

a) Για το πρόγραμμα υπάρχει ένα αρχείο *makefile*. Η μεταγλώττιση του προγράμματος γίνεται με την εντολή:

`$ make all`

Για να τρέξει το πρόγραμμα πρέπει να δοθεί η εντολή:

`$ ./quic $(FLAGS) $(SOURCE) $(DESTINATION)`

όπου το *source* είναι το *directory* του οποίου τα αρχεία θέλουμε να αντιγράψουμε και *destination* το *directory* στο οποίο θέλουμε να αντιγραφούν τα αρχεία. Τα *flags* μπορεί να είναι:

- `-v` , το οποίο ενεργοποιεί την εκτύπωση των μηνυμάτων όπως: “Created /.../” και “Deleted /.../” .
- `-d` , το οποίο ενεργοποιεί την λειτουργία κατά την οποία αρχεία τα οποία έχουν διαγραφεί από το *source directory*, δεν πρέπει να υπάρχουν στο *destination directory* κατά την πραγματοποίηση της μεταφοράς των αρχείων.

b) Η εργασία υλοποιήθηκε σε C++ καθώς επέλεξα να αξιοποιήσω ορισμένες λειτουργίες της όπως την ύπαρξη της κλάσης *string*.

c) Η εργασία αποτελείται από τα εξής αρχεία:

- **quic.cpp**, το οποίο περιέχει την *main* και εκτελεί τον βασικό κορμό λειτουργιών του προγράμματος.
- **quic.hpp**, το οποίο περιέχει τις βιβλιοθήκες του *quic.cpp* καθώς και ορισμένες *global variables* οι οποίες κυρίως χρησιμεύουν στην εκτύπωση των αποτελεσμάτων, όπως ακριβώς ορίζονται στην εκφώνηση.
- **copyFunctions.cpp**, το οποίο περιέχει τις συναρτήσεις που χρησιμεύουν κυρίως στην αντιγραφή των αρχείων στα *directories*.
- **copyFunctions.hpp**, το οποίο περιέχει τις βιβλιοθήκες του *copyFunctions.cpp*.
- **deleteFunctions.cpp**, το οποίο περιέχει τις συναρτήσεις που χρησιμεύουν κυρίως στην διαγραφή των αρχείων από τα *directories*.
- **deleteFunctions.hpp**, το οποίο περιέχει τις βιβλιοθήκες του *deleteFunctions.cpp*.

d) Στην εργασία, επέλεξα να μην υλοποιήσω το *flag* “-l” το οποίο ενεργοποιεί την διαχείριση των *soft/hard links* σύμφωνα με τις προδιαγραφές τους ως τύπος αρχείου. Παρόλα αυτά, τα *soft/hard links* αντιμετωπίζονται επιτυχώς ως κανονικά αρχεία.

e) Κατά την εκκίνηση του προγράμματος, εάν έχει δοθεί σαν όρισμα το *flag* “-d” εκτελείται η συνάρτηση `::checkToDelete` η οποία ελέγχει ουσιαστικά για αρχεία τα οποία υπάρχουν στο *destination directory* και όχι στο *source directory* και όποιο βρεί το διαγράφει. Επέλεξα την συγκεκριμένη υλοποίηση καθώς θεώρησα πιο σωστό το να ξεμπερδεύει το πρόγραμμα από τέτοιου είδους αρχεία εξ αρχής ώστε στην υπόλοιπη εκτέλεση να έχει να κάνει μόνο με “όμοια” ή ελλιπή αρχεία.

f) Κατά την αντιγραφή των αρχείων, το πρόγραμμα ελέγχει κάθε αρχείο στο *source directory*, λαμβάνει τα κατάλληλα μέτρα εάν είναι *directory* ( αναδρομή ), και εάν το πρόγραμμα βεβαιωθεί πως το αρχείο δεν υπάρχει στον *destination directory* ( κριτήρια όμοιων αρχείων ) τότε το αντιγράφει στον φάκελο.

g) Η αντιγραφή των αρχείων γίνεται με *deep copy*, δηλαδή αντιγράφεται το περιεχόμενο του αρχείου χαρακτήρα προς χαρακτήρα και τέλος δίνονται τα *permissions* του αρχικού αρχείου στο νέο. Επέλεξα να αλλάζω μόνο το περιεχόμενο των αρχείων και τα *permissions* καθώς η ημερομηνία τελευταίας τροποποίησης πρέπει να διατηρηθεί ως έχειν για περαιτέρω λειτουργίες του προγράμματος, και το μέγεθος με τη συγκεκριμένη μέθοδο αντιγραφής μεταφέρεται στο νέο αρχείο με σωστό τρόπο. Τα υπόλοιπα χαρακτηριστικά του αρχείου δεν πειράζονται, σύμφωνα πάντα με την εκφώνηση της άσκησης και τις υποδείξεις στο *riazza*.

h) Επέλεξα να αγνοώ σε διάφορα σημεία του προγράμματος τα αρχεία τύπου “./” και “../” τα οποία αποτελούν δείκτες για κάθε *directory* στο προηγούμενο φάκελο του *path* και στον αρχικό φάκελο της ιεραρχίας αντίστοιχα, καθώς τα συγκεκριμένα αρχεία δημιουργούνται αυτόματα για κάθε φάκελο και επίσης μπορεί να οδηγήσουν μια εκτέλεση σε ατέρμονο *loop*.

i) Η συνάρτηση *::checkCriteria* ελέγχει τα κριτήρια για το αν δύο αρχεία είναι όμοια ή όχι, σύμφωνα πάντα με τις προδιαγραφές που αναφέρονται στην εκφώνηση της άσκησης.

j) Κατά τον τερματισμό του προγράμματος απελευθερώνεται κάθε *block* που έχει γίνει *alloced* κατά την εκτέλεση.

**Σας εύχομαι καλή διόρθωση!!**