**ΑΝΑΦΟΡΑ ΕΡΓΑΣΙΑΣ ΣΤΑ ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ 1**



ΘΕΣΣΑΛΟΝΙΚΗ , ΑΠΡΙΛΙΟΣ 2021

**ΚΕΦΑΛΑΙΟ 1:**

Σχολιασμός του κώδικα και ανάλυση των προγραμματιστικών τεχνικών

Αρχικά, αξίζει να αναφερθεί πως η τελική μορφή του κώδικά μου είναι ομολογουμένως κάπως περίεργη , καθώς έχουν υλοποιηθεί όλα τα ζητούμενα της εργασίας ουσιαστικά σε μια μόνο συνάρτηση , με τις computeber() και computeXOR() να είναι συμπληρωματικές και να επιτελούν πολύ απλές λειτουργίες που θα αναλυθούν στην συνέχεια. Η μορφή του κώδικα αυτή είναι απόρροια της ενασχόλησής μου με κάθε υποερώτημα της εργασίας ξεχωριστά, δίχως να γνωρίζω εξ’ αρχής πως πολλά από αυτά μοιράζονται κοινές προγραμματιστικές υλοποιήσεις που θα μπορούσαν να δομήσουν ξεχωριστές συναρτήσεις. Όταν πλέον ολοκλήρωσα όλα τα επιμέρους τμήματα του κώδικά μου και συνειδητοποίησα αυτόν τον συσχετισμό , θα ήταν χρονοβόρο το να οργανώσω ξανά από την αρχή τα πάντα και για αυτό επέλεξα να επιμείνω στην αρχική μου υλοποίηση.

Περνώντας τώρα στον κώδικα αυτό καθαυτό , μπορούμε ξεκάθαρα να εντοπίσουμε τον καταλυτικό ρόλο που παίζει το πρότυπο του αντικειμένου τύπου modem που ήταν διαθέσιμο στην Ithaki. Το συγκεκριμένο κομμάτι κώδικα έχει χρησιμοποιηθεί σε όλα τα υποπροβλήματα της συγκεκριμένης , έτσι ώστε μέσω της ρουτίνας επανάληψης for(;;) να μπορούμε να λαμβάνουμε ένα προς ένα τα byte που στέλνει ο server του μαθήματος. Φυσικά, η υπορουτίνα αυτή έχει εμπλουτιστεί με επιπλέον συνθήκες τερματισμού , ούτως ώστε να αναγνωρίζεται από τον χρήστη η επιτυχημένη μεταβίβαση ενός πακέτου μηνυμάτων από τον server και να είναι δυνατή η άμεση παραλαβή του επόμενου πακέτου , χωρίς να πρέπει να μεσολαβήσει το timeout που έχουμε ορίσει για το modem της ιθάκης. Συμπληρωματικά , στα ερωτήματα που απαιτείται η αποθήκευση των byte που λαμβάνουμε σε μορφή jpeg η λούπα ανάγνωσης που περιγράψαμε νωρίτερα ελέγχει συνεχώς την ακολουθία των λαμβανόμενων στοιχείων και τερματίζει εφόσον εντοπίσει τα δύο διαδοχικά byte τερματισμού αρχείου εικόνας (217 και 255 στο δεκαδικό σύστημα). Ειδικότερα, στην αποθήκευση εικόνων μέσω του προγραμματιστικού περιβάλλοντος της java που χρησιμοποιούμε μας βοηθάει και η κλάση java.nio.file.Files σε συνδυασμό με την java.nio.file.Path που δημιουργεί ένα «μονοπάτι» μνήμης στο οποίο αποθηκεύονται οι πίνακες τύπου byte[] που δημιουργούμε με τα δεδομένα εισόδου.

Στην συνέχεια, όσον αφορά την αποθήκευση των δεδομένων εισόδου στο excel για την μετέπειτα μετουσίωσή τους σε διαγράμματα χρησιμοποιούμε την κλάση apache poi της java , μέσω της οποίας μεταβιβάζουμε σε κελιά που δημιουργούμε τις επιθυμητές τιμές. Παράλληλα , για τον χρονικό έλεγχο του προγράμματος χρησιμοποιούνται η κλάση instant και η μέθοδος currentTimeMillis() που επιστρέφουν την χρονική στιγμή κατά την οποία καλούνται.

Τέλος , ας περάσουμε στην ανάλυση του πιο σύνθετου υποτμήματος του κώδικα το οποίο είναι ο μηχανισμός ARQ που χρησιμοποιείται για την διόρθωση σφαλμάτων που παράγει ο server σε ακολουθίες 16 byte. Το πρόγραμμα που δημιουργήσαμε χρησιμοποιεί έναν δείκτη που εντοπίζει τις θέσεις των bit από τα εισερχόμενα πακέτα που πρέπει να αποθηκεύσουμε και να υπολογίσουμε το xor τους μέσω της μεθόδου computeXOR(). Έπειτα , η μέθοδος computeBER() υπολογίζει την σταθερά ber μέσω του τύπου (ack)/(ack+nack) = (1 – ber)^l , όπου τα ack και nack είναι οι συνολικές μεταβιβάσεις κωδικών ack και nack αντίστοιχα στον server και το l το μήκος της ακολουθίας των bit. Επομένως , το l ισούται με 128 , λόγω της αποθήκευσης 16 byte ανά πακέτο , το καθένα εκ των οποίων αποτελείται όπως είναι γνωστό από 8 bit. Συμπληρωματικά , το τελευταίο στάδιο υλοποίησης του ARQ είναι η σύγκριση του xor της ακολουθίας των byte με τον τριψήφιο κωδικό fcs και η μέτρηση των επανεκομπών του κάθε πακέτου , έως ότου το σφάλμα που είχε να διορθωθεί.

**ΚΕΦΑΛΑΙΟ 2:**

Σχολιασμός και παρατηρήσεις για τα ζητούμενα διαγράμματα

Τα γραφήματα G1 και G2 έχουν ως ζητούμενο την αποτύπωση της χρονικής απόκρισης του κάθε πακέτου που καταφτάνει από τον server , το πρώτο για τα echo packets και το δεύτερο για τα πακέτα που ελέγχονται από τον μηχανισμό ARQ. Μπορεί κανείς πολύ εύκολα να παρατηρήσει πως οι τιμές του χρόνου απόκρισης κινούνται γύρω από τα 50 ms με εξαίρεση ορισμένες στιγμιαίες εκρήξεις του server.

Από την άλλη πλευρά , το διάγραμμα G3 αποτελεί μια αναπαράσταση σε μορφή ραβδογράμματος του πλήθους των επανεκπομπών του κάθε πακέτου που λαμβάνει ο χρήστης μέσω του μηχανισμού ARQ. Το διάγραμμα αυτό καθιστά λοιπόν σαφές πως η κατανομή πιθανότητας του αριθμού επανεκπομπών του κάθε πακέτου μπορεί εύκολα να προσεγγιστεί μέσω της γεωμετρικής κατανομής , καθώς όπως είναι γνωστό αυτό το είδος κατανομής χρησιμοποιείται για την καταμέτρηση των αποτυχιών ανεξαρτήτων μεταξύ τους πειραμάτων Bernoulli , μέχρι την πρώτη επιτυχία.

**ΚΕΦΑΛΑΙΟ 3:**

Μηχανισμοί και πρωτόκολλα λειτουργίας διατάξεων modem

DHCP:

Το DHCP (*Dynamic Host Configuration Protocol*) αποτελεί έναν μηχανισμό διαχείρισης πρωτοκόλλων TCP/IP. Η διαφορά αυτού του πρωτόκολλου είναι πως , εκτός από την στατική IP διεύθυνση της συσκευής του, αποδίδει σε έναν χρήστη και δυναμικές διευθύνσεις IP που αλλάζουν ανάλογα με το δίκτυο στο οποίο είναι συνδεδεμένος. Το DCHP υποστηρίζει 3 μηχανισμούς για να αντιστοιχίζει διευθύνσεις. Αυτοί είναι:

1. Αυτόματη αντιστοίχηση (με αντιστοίχηση μόνιμης διεύθυνσης)
2. Δυναμική αντιστοίχηση (με διεύθυνση με ημερομηνία λήξης)
3. Χειροκίνητη αντιστοίχηση (ο διαχειριστής κανονίζει ότι θεωρεί καλύτερο)

NAT:

Το ΝΑΤ (Network Address Translation) είναι ειδικό πρωτόκολλο που εκτελούν οι πύλες (gateways) και έχει σαν αποτέλεσμα να αλλάζει την IP διεύθυνση ενός πακέτου που ξεκινά από έναν υπολογιστή εντός του τοπικού δικτύου και προωθείται εκτός του δικτύου.

ΑRP:

O τρόπος που λειτουργεί το πρωτόκολλο ARP είναι εξαιρετικά απλός. Κάθε φορά που πρέπει να γίνει γνωστή η διεύθυνση MAC που αντιστοιχεί σε κάποια συγκεκριμένη διεύθυνση IP , λαμβάνει χώρα εκπομπή σε όλους τους υπολογιστές ενός πακέτου δεδομένων που περιέχει τη διεύθυνση IP που θέλουμε να μεταφράσουμε . Ο κάθε ένας από τους υπολογιστές του δικτύου , παραλαμβάνει αυτό το πακέτο , συγκρίνει τη διεύθυνση IP που περιέχει , με τη δική του διεύθυνση IP και εάν οι δύο διευθύνσεις είναι οι ίδιες , αποστέλλει μια απάντηση στον υπολογιστή που υπέβαλλε το ερώτημα . H απάντηση αυτή περιέχει τη MAC διεύθυνση του υπολογιστή αποστολέα η οποία ταυτοποιείται , απομονώνεται και αποθηκεύεται σε μια ειδική μνήμη ARP cache , έτσι ώστε να μπορεί να χρησιμοποιηθεί στο μέλλον.