Ανάπτυξη Λογισμικού Για Πληροφοριακά Συστήματα
Μέρος 1ο

Όνομα : Δημάκης Βασίλειος , Δεωνάς Παύλος
Α.Μ : 1115201400324 , 1115201500033

Για την ταξινόμηση των σχέσεων ακολουθήσαμε τον τρόπο που περιγράφηκε στο μάθημα. Δηλαδή παίρνουμε την καθε σχέση της οποιας τις τιμες κραταμε σε εναν πινακα ,ταξινομούμε για το πρώτο byte με βαση το key καθε σχεσης και τα γράφουμε σε έναν αλλο πίνακα κρατώντας δείκτες στον πινακα για το που ξεκινάνε και για το που τελειώνουνε οι εγγραφές που σαν 1ο byte εχουν 0,1,2,3 κ.ο.κ. Στην συνέχεια ταξινομούμε με βαση το 2ο byte και περναμε τις τιμές στον 1ο πίνακα κρατώντας παλι δείκτες οπως παραπάνω.Αυτο γινεται μεχρι η καθε ομάδα εγγραφών (ομαδα εννοω σχετικα με τους δεικτες που κραταμε πανω στον πινακα) να περιέχει λιγότερες εγγραφές απο 64KB .Όταν φτασουμε σε αυτο το σημείο καλουμε την συναρτηση quicksort (περιγράφεται παρακάτω) για καθε ομαδα και τοτε εχουν ταξινομηθει ολες οι εγγραφες.
Ολη αυτη η διαδικασία υλοποιείται με την συναρτηση : `void smj(relation * array0, relation * array1 ,int start,int end,int where_to_write,int byte)`

Για την εύρεση των τελικών αποτελεσμάτων έχει φτιαχτεί μια συνάρτηση FindRes η οποία παιρνει σαν ορισματα τους δυο μας ταξινομημένους πίνακες. Η διαδικασια που ακολουθεί ειναι αρχικα να διατρεχει τον εναν απο τους δυο πινακες (τον πιο μικρο σε μεγεθος -απο επιλογη-) και να προχωραει στον αλλο μεχρι να βρει καποιο match. Αν βρεθει, γινεται εισαγωγη στην λιστα αυξανεται ενας μετρητης matches και ενας ακομα num_of_matches και προχωραει ο δεικτης μας στο επομενο στοιχειο του δευτερου πινακα. Ο μετρητης matches χρησιμοποιειται για να μπορουμε στην περιπτωση που εχουμε διπλοτυπα στον πινακα τον οποιο εχουμε επιλεξει να διατρεξουμε, να ξερουμε ακριβως που θα βρουμε τα αποτελεσματα μας και να μην χρειαζεται καθε φορα να διατρεχουμε και τον δευτερο πινακα για να βρισκουμε αποτελεσματα. Επομενως το matches μηδενιζεται καθε φορα που αλλαζει το στοιχειο απο τον πρωτο πινακα.
Το num_of_matches ειναι ενας μετρητης ο οποιος μας κραταει τα συνολικα joins που γινονται.
Η λιστα τωρα ειναι μια απλη λιστα οπου κραταμε δυο δεικτες στην αρχη της και στο τελος της. Σε καθε κομβο της λιστας κραταμε τα payloads που μας ενδιαφερουν ανα δυαδες, ακριβως οπως γινονται τα matches και εναν δεικτη για τον επομενο κομβο. Η εισαγωγη στη λιστα γινεται μεχρι να γεμισει καθε κομβος, δηλαδη να φτασει το 1MB και σε περιπτωση που πρεπει να φτιαχτει καποιος καινουργιος κομβος για να κερδισουμε χρονο αξιοποιησαμε τον δεικτη της λιστας που δειχνει καθε φορα στο τελος της. Ετσι δεν χρειαζοταν να διατρεχουμε ολη την λιστα και ηταν πιο ευκολο και γρηγορο το να δημιουργουμε κομβους και να κανουμε την εισαγωγη των αποτελεσματων μας.
Υπαρχουν και συναρτησεις εκτυπωσεις της λιστας αλλα και απελευθερωσης της απο τη μνημη. (PrintResults(), freelist())

Για την ταξινομηση με quicksort εχει δημιουργηθει μια συναρτηση quicksort μια συναρτηση partition και μια συναρτηση swap. Στις συναρτησεις quicksort και partition στελνουμε τον πινακα μας μαζι με δυο αριθμους low , high, που χρησιμοποιουμε για να ξερουμε ποια στοιχεια του πινακα πρεπει να ταξινομησουμε.

Τα αποτελέσματα του join βρίσκονται σε ενα αρχείο που ονομάζεται Results.csv που εχουμε δημιουργήσει και περνάμε τα αποτελέσματα εκει.
Επίσης γίνεται μια εκτύπωση στο output σχετικα με τον αριθμό των joins.

Για την μεταγλώττιση του προγράμματος εχει υλοποιηθεί κατάλληλο Makefile οποτε αρκεί η εντολή : make

Επίσης με make clean διαγράφονται τα object αρχεία αλλα και το αρχείο Results.csv που περιέχει τα αποτελέσματα του join.

Για την εκτέλεση του προγράμματος αρκει η εντολή : ./main <file1> <file2>
,οπου file1 η πρωτη σχέση και file2 η δευτερη ωστε να γινει το join μεταξυ αυτών των 2.