

ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2017-2018

3Η ΕΡΓΑΣΙΑ

ΣΥΜΜΕΤΕΧΟΝΤΕΣ ΣΤΗΝ ΟΜΑΔΑ:

ΕΥΑΓΓΕΛΙΟΥ ΠΑΝΑΓΙΩΤΗΣ
ΣΠΙΘΑΣ ΕΥΑΓΓΕΛΟΣ

AM:1115201500039
AM:1115201500147

Εντολή για μεταγλώττιση και των 3 sr_main συναρτήσεων:

- make

Εντολές για εκτέλεση:

- sr_main1 -> ./build/sr_main1
- sr_main2 -> ./build/sr_main2
- sr_main3 -> ./build/sr_main3

Για διαγραφή των αρχείων που δημιουργήθηκαν καθώς και των εκτελέσιμων προγραμμάτων τρέχουμε την εντολή **make clean**.

Το πρόγραμμα τρέχει σωστά και έχουν υλοποιηθεί όλα όσα αναφέρονται στην εκφώνηση. Σύμφωνα με τον valgrind δεν υπάρχει κανένα error και κανένα memory leak (εκτός από το leak που λέει ότι ένα block είναι still reachable και το οποίο αναφέραμε και στο e-class στην εργασία 1).

Συναρτήσεις Επιπέδου Ευρετηρίου SR (Sorted File)

Όλες οι συναρτήσεις εκτός της SR_SortedFile είναι ίδιες από την εργασία 1.

SR_ErrorCode SR_SortedFile:

- Αντιγράφει το inputFile σε ένα tempFile το οποίο στην συνέχεια δίνεται στην inplace_Sorting η οποία θα κάνει το Βήμα 0 του αλγορίθμου εξωτερικής ταξινόμησης.
- Πιο συγκεκριμένα:
 - Αυτή η συνάρτηση θα καθορίζει κάθε φορά ποια μπλοκ θα ταξινομηθούν.
 - Αρχικά ξεκινάει με το πρώτο μπλοκ μέχρι το μπλοκ buffer size και στη συνέχεια ταξινομεί τα επόμενα γκρουπς μεγέθους buffer size.
 - Η ταξινόμηση γίνεται με τη συνάρτηση Quick_Sort.
 - Αυτή η συνάρτηση χρησιμοποιεί την Partition για να βρει την οριστική θέση του πρώτου στοιχείου και να το βάλει εκεί και στη συνέχεια ταξινομεί όλα τα στοιχεία δεξιά από την οριστική θέση του και αριστερά.
 - Η partition αυτό που κάνει είναι να παίρνει το πρώτο στοιχείο κάθε φορά και να το συγκρίνει με το όλα τα στοιχεία μέχρι το right.
 - Κάθε φορά που βρίσκει κάποιο στοιχείο μικρότερο ή ίσο του πρώτου το αντιμεταθέτει με το επόμενο αυτού που αντιμετάθεσε τελευταίο.
 - Η πρώτη αντιμετάθεση θα γίνει με το δεύτερο στοιχείο μετά το πρώτο.

- Στο τέλος θα αντιμεταθέσει το πρώτο στοιχείο με το τελευταίο στοιχείο που αντιμετάθεσε. Για την ταξινόμηση όλα τα μπλοκ, κάθε γκρουπ θεωρούνται σαν ένας μεγάλος πίνακας με συνεχόμενα στοιχεία.
- Γίνονται οι κατάλληλες ενεργειες σε κάθε συνάρτηση για να επιτευχθεί αυτό.
- Πλέον το tempFile θα έχει ταξινομημένα τμήματα μεγέθους bufferSize block το καθένα, το οποίο και δίνουμε στην mergeSort η οποία θα κάνει τα υπόλοιπα βήματα του αλγορίθμου.
- Πιο συγκεκριμένα:
 - Αρχικά δημιουργεί και ένα δεύτερο tempFile το οποίο θα γίνεται αλλαγή κάθε φορά με το άλλο tempFile ώστε το ένα να αποτελεί το input (δηλαδή την παλιά κατάσταση) και το άλλο να αποτελεί το output (δηλαδή την νέα κατάσταση) . Έτσι στο τέλος κάθε επανάληψης το output γίνεται το νέο input και το input “καθαρίζεται” ώστε να γίνει το νέο output. Με αυτό τον τρόπο επιτυγχάνουμε να χρησιμοποιούμε μόνο 2 temp αρχεία.
 - Στην αρχή επίσης υπολογίζουμε και τον αριθμό των ταξινομημένων groups.
 - Στην συνέχεια κάνουμε ένα while loop μέχρι να καταλήξουμε να έχουμε μόνο 1 group που θα είναι και το τελικό ταξινομημένο μας group.
 - Σε κάθε επανάληψη θα γίνουν merges ανά bufferSize-1 groups και έτσι καθορίζουμε το αρχικό block, το τελικό block καθώς και κάποιες άλλες πληροφορίες για την κάθε merge και καλούμε την συνάρτηση mergeGroups.
 - Η συνάρτηση αυτή θα αποθηκεύσει σε έναν πίνακα τύπου groupInfo πληροφορίες για το κάθε group.
 - Επίσης έχουμε και έναν πίνακα τύπου inputDataBlocks όπου αναπαριστά τους input buffers που φορτώνονται και έχουν data από κάποιο block κάποιου group.
 - Αρχικά ο “δείκτης” firstGroup δείχνει στο αριστερότερο group της current merge και ο lastGroup στο δεξιότερο group.
 - Γίνεται ένα while loop μέχρι το firstGroup == lastGroup , δηλαδή μέχρι να έχει μείνει μόνο 1 group (οπότε και δε θα υπάρχει νόημα να συγκριθεί με κάποιο άλλο group, αφού δε θα υπάρχουν άλλα groups).
 - Έτσι σε κάθε επανάληψη βρίσκουμε την μικρότερη record από κάποιο group και την γράφουμε στον output buffer (outDataBlock) .
 - Για να το κάνουμε αυτό αρχικά στον πίνακα groupsArray , που έχει τις πληροφορίες για κάθε group, φορτώνουμε την μικρότερη εγγραφή από κάθε group ώστε να μην χρειαστεί κάθε φορά να φορτώνουμε την ίδια εγγραφή.
 - Έτσι συγκρίνουμε τις εγγραφές που έχουμε φορτώσει στον πίνακα groupsArray και όταν βρούμε την μικρότερη, την γράφουμε στο outDataBlock και παίρνουμε την επόμενη εγγραφή από το συγκεκριμένο group.
 - Όταν λοιπόν τελειώσει το while loop έχουμε μείνει πλέον με μόνο 1 group, οπότε και γράφουμε τα υπόλοιπα records του συγκεκριμένου group όπως είναι στο tempOutputFile.
 - Τέλος κάνουμε copy το tempOutputFile στο κανονικό outputFile και διαγράφουμε τα temp αρχεία.