

Project 3 - External Sort

Συντελεστές :

ΕΠΙΘΕΤΟ : ΚΑΡΑΜΠΑΣ

ΟΝΟΜΑ : ΕΛΕΥΘΕΡΙΟΣ - ΑΡΓΥΡΙΟΣ

ΑΜ : 1115201400064

ΕΠΙΘΕΤΟ : ΚΟΥΚΟΥΛΗΣ

ΟΝΟΜΑ: ΑΘΑΝΑΣΙΟΣ

ΑΜ: 1115201400328

Για την υλοποίηση της εργασίας ακολουθήθηκε η εξής δομή για τα αρχεία :

- 1) **include** : Περιέχει όλα τα αρχεία κεφαλίδας που χρειάζονται .

Φάκελος include	
<u>Όνομα Αρχείου</u>	<u>Περιγραφή</u>
Compare.h	περιέχει την δήλωση της συνάρτησης Compare
Print.h	περιέχει την δήλωση της συνάρτησης PrintBlock
QuickSort.h	περιέχει την δήλωση της συνάρτησης QuickSort καθώς και των βοηθητικών της
Sort.h	περιέχει την δήλωση της συνάρτησης
Struct.h	Περιέχει την δήλωση της δομής Block_Data , η οποία χρησιμοποιείται στην Merge Sort για να χειριζόμαστε ταυτόχρονα τα πολλαπλά μπλοκ όταν κάνουμε την ταξινόμηση.
bf.h	-----
sort_file.h	-----

2) **src** : Περιέχει τον βασικό κορμό της άσκησης .

Φάκελος src	
Όνομα Αρχείου	Περιγραφή
<i>Compare.c</i>	Περιέχει την συνάρτηση <i>compare</i> που χρησιμοποιείται για την σύγκριση δύο <i>Records</i>
<i>Print.c</i>	Περιέχει την συνάρτηση <i>PrintBlock</i> η οποία χρησιμοποιείται από την συνάρτηση <i>SR_PrintAllEntries</i> για την εκτύπωση των εγγραφών ενός μπλοκ.
<i>QuickSort.c</i>	Περιέχει τις συναρτήσεις που χρησιμοποιούνται στον αλγόριθμο της Quicksort.
<i>Sort.c</i>	Περιέχει την υλοποίηση της <i>Merge Sort</i> καθώς και των βοηθητικών συναρτήσεων που αυτή χρησιμοποιεί .
<i>sort_file.c</i>	Υλοποιούνται οι βασικές συναρτήσεις της <i>SR_SortedFile</i> .

Στον παρακάτω πίνακα παρατίθενται οι βασικές συναρτήσεις της άσκησης :

Αρχείο <i>sort_file.c</i> - βασικές συναρτήσεις	
<u>Όνομα Συνάρτησης</u>	<u>Περιγραφή</u>
<i>SR_Init</i>	Αρχικοποιεί τον επίπεδο <i>SR</i> .
<i>SR_CreateFile</i>	Αρχικοποιούμε το πρώτο μπλοκ εισάγοντας στα πρώτα byte το αναγνωριστικό “SR” ώστε να γνωρίζουμε οτι πρόκειται για αρχείο ταξινόμησης .
<i>SR_OpenFile</i>	Ανοίγει το αρχείο και ελέγχει εαν είναι αρχείο ταξινόμησης.
<i>SR_CloseFile</i>	Κλείνει το αρχείο ταξινόμησης .
<i>SR_InsertEntry</i>	Εισάγουμε τα δεδομένα από την δομή <i>Record</i> που

	λαμβάνουμε σαν όρισμα στην τελευταία θέση του αρχείου ταξινόμησης .
SR_SortedFile	Ανοίγουμε το αρχείο εισόδου και δημιουργούμε το αρχείο εξόδου . Έπειτα καλούμε τη συνάρτηση ταξινόμησης ώστε να ταξινομηθούν κατάλληλα τα περιεχόμενα του αρχείου εισόδου και τα τοποθετούμε στο αρχείο εξόδου .
SR_PrintAllEntries	Εκτυπώνει όλες τις εγγραφές που βρίσκονται στο αρχείο ταξινόμησης .

Στον παρακάτω πίνακα παρατίθενται οι βοηθητικές συναρτήσεις της άσκησης:

Αρχείο QuickSort.c - Βοηθητικές συναρτήσεις	
<u>Όνομα Συνάρτησης</u>	<u>Περιγραφή</u>
QuickSort	Υλοποιεί τον αναδρομικό αλγόριθμο της Quicksort .
Swap	Βοηθητική συνάρτηση της Quicksort , μεταθέτει τις τιμές .
Partition	Βοηθητική συνάρτηση της Quicksort , ώστε να γίνει διαμέριση .

Αρχείο Print.c - Βοηθητικές συναρτήσεις	
<u>Όνομα Συνάρτησης</u>	<u>Περιγραφή</u>
PrintBlock	Βοηθητική συνάρτηση της SR_PrintAllEntries . Δέχεται ως όρισμα έναν αριθμό μπλοκ και εκτυπώνει τα δεδομένα του .

Αρχείο Sort.c - Βοηθητικές συναρτήσεις	
<u>Όνομα Συνάρτησης</u>	<u>Περιγραφή</u>
SortBlock	Αρχικοποιεί τα δεδομένα και καλεί την Quicksort .
Initialize_BlockData	Αρχικοποιεί την δομή Block_Data . Στην περίπτωση που είμαστε στο βήμα 0 κάνει εσωτερική ταξινόμηση στο μπλοκ .
Find_Min	Επιστρέφει την θέση που ανήκει το μπλοκ, με την μικρότερη τρέχουσα τιμή, στον πίνακα της δομής Block_Data ώστε να το επιλέξει η Merge Sort .
MergeSort	Δημιουργεί ένα βοηθητικό temp αρχείο ώστε να υλοποιηθεί η ταξινόμηση . Δημιουργεί την δομή Block_Data . Βρίσκει σε ποιο πεδίο θα γίνει η ταξινόμηση και αρχικοποιεί κατάλληλα τις μεταβλητές που θα χρειαστούν . Έπειτα βρίσκουμε τον αριθμό των επαναλήψεων που θα χρειαστούν για την ταξινόμηση (με βάση τα διαθέσιμα μπλοκ της μνήμης) και υπολογίζουμε με ποιο αρχείο πρέπει να ξεκινήσουμε ώστε στο τέλος το ταξινομημένο αρχείο να είναι το αρχείο εξόδου μας και όχι το βοηθητικό (temp) . Τέλος υλοποιείται ο αλγόριθμος της Merge Sort χρησιμοποιώντας τις προαναφερθείσες βοηθητικές συναρτήσεις .

Αρχείο Compare.c - Βοηθητικές συναρτήσεις	
<u>Όνομα Συνάρτησης</u>	<u>Περιγραφή</u>
Compare.c	Η συνάρτηση αυτή είναι η ίδια που έχουμε υλοποιήσει στην 2η εργασία (B+ tree) και παίρνει δυο στοιχεία , βρίσκει τον τύπο τους και τα συγκρίνει .