

Ηλίας Μεντζέλος – 1115201400106
Ευάγγελος Πέκος – 1115201400157
Ευφροσύνη Παντελιάδη – 1115201400301

Η υλοποίηση της εργασίας πραγματοποιήθηκε στη γλώσσα C και βρίσκεται στο φάκελο `heap_file_64`. Για τον σκοπό αυτό, δημιουργήθηκε ένα πλήθος βοηθητικών συναρτήσεων και δομών, που συμβάλλουν στη λειτουργία των τμημάτων κώδικα που ζητήθηκε να συμπληρωθούν. Παρακάτω δίνεται μια μικρή περιγραφή, που αφορά τη δομή της εργασίας, προκειμένου να γίνει ευκολότερη η κατανόησή της.

Δομή Blocks:

Η δομή ενός block ευρετηρίου είναι:

“E”	counter	P ₀	K ₁	P ₁	K ₂	...	K _n	P _n
-----	---------	----------------	----------------	----------------	----------------	-----	----------------	----------------

- Το string “E” στα πρώτα `sizeof(“E”)` bytes του block, φανερώνει πως πρόκειται για block ευρετηρίου.
- Ο counter αποτελεί μια ακέραια τιμή, που φανερώνει το πλήθος των κλειδιών στο συγκεκριμένο block. Η τιμή του μας βοηθά να διατρέχουμε πιο γρήγορα το block, καθώς και να βρίσκουμε αμέσως εάν υπάρχει διαθέσιμος χώρος ή όχι.
- Με P_i συμβολίζουμε τους δείκτες (δηλαδή `block_num`) σε block ευρετηρίου/δεδομένων.
- Με K_i συμβολίζουμε τις τιμές των κλειδιών.

Η δομή ενός block δεδομένων είναι:

“D”	P _{next}	counter	Record ₁	...	Record _n
-----	-------------------	---------	---------------------	-----	---------------------

- Το string “D” στα πρώτα `sizeof(“D”)` bytes του block φανερώνει πως πρόκειται για block δεδομένων.
- Ο P_{next} αποτελεί ένα δείκτη (δηλαδή `block_num`) στο αμέσως επόμενο (γειτονικό) block δεδομένων.
- Ο counter αποτελεί μία ακέραια τιμή, που φανερώνει το πλήθος των εγγραφών στο συγκεκριμένο block. Η τιμή του μας βοηθά να διατρέχουμε πιο γρήγορα το block, καθώς και να βρίσκουμε αμέσως εάν υπάρχει διαθέσιμος χώρος ή όχι.
- Με Record_i συμβολίζουμε τις εγγραφές, οι οποίες αποτελούνται από δύο πεδία συγκεκριμένου μήκους.

Δομές:

Οι δομές δεδομένων που δημιουργήθηκαν είναι:

- `Open_File`: Έχει ως μέλη της πληροφορίες που αφορούν ένα ανοιχτό αρχείο, όπως το όνομά του, το είδος και το μήκος του κάθε πεδίου μιας εγγραφής του κτλ.
- `scan_point`: Κρατάμε το σημείο που βρίσκεται το `scan` έχοντας το `value1`, που προορίζεται για σύγκριση και το `value2`, που είναι για εκτύπωση/επιστροφή. Επιπλέον στην συγκεκριμένη δομή φυλάσσονται στοιχεία που μπορούμε να βρούμε για την συγκεκριμένη εγγραφή (π.χ σε ποιο

block βρίσκεται, ποιο είναι το επόμενο του block, το πλήθος των εγγραφών του block και την σειρά της συγκεκριμένης εγγραφής στο block)

- Scan: Έχει ως μέλη της πληροφορίες που αφορούν τα αρχεία που σαρώνονται.

Στοίβα:

Δημιουργήθηκε για να μπορούμε να φυλάμε τους γονείς των blocks.

Για την υλοποίησή της δημιουργήθηκαν ορισμένες δομές δεδομένων (stack_node, stack), καθώς και συναρτήσεις (create_stack, create_node, push, pop, clean_stack) που αφορούν τη λειτουργία της.

Βοηθητικές Συναρτήσεις:

Οι συναρτήσεις που δημιουργήθηκαν προκειμένου να επιτευχθεί η λειτουργία της άσκησης, καθώς και η οργάνωση του κώδικα είναι:

- write_value: Γράφει μία τιμή value τύπου attr και μήκους length στην περιοχή μνήμης που δείχνει ο δείκτης data.
- read_value: Διαβάζει μία τιμή
- read_int_value: Διαβάζει μία ακέραια τιμή
- checkBF: Εκτυπώνει το κατάλληλο μήνυμα σφάλματος και τερματίζει την εκτέλεση, εφόσον αυτό χρειάζεται
- CompareKeys: Συγκρίνει δύο κλειδιά τύπου type επιστρέφοντας τη διαφορά τους
- hashfile: Συνάρτηση κατακερματισμού που επιστρέφει μία ακέραια τιμή, η οποία αποτελεί το υπόλοιπο της διαίρεσης του file descriptor ενός αρχείου με την τιμή MAXFILES
- getChildBlock: Διατρέχει ένα block ευρετηρίου και επιστρέφει το κατάλληλο block_num του block που θα πρέπει να διατρέξουμε στην συνέχεια.
- writeNums: Βρίσκει το block_num που θα πρέπει να αναθέσουμε ως γείτονα στο block που δεσμεύθηκε πρόσφατα και αλλάζει στο block ευρετηρίου το $P_i = -1$ με το block_num του καινούριου block.
- insert_AvailableSpace: Τοποθετεί μία νέα εγγραφή σε ένα block δεδομένων που έχει διαθέσιμο χώρο.
- get_most_left: Επιστρέφει το αριστερότερο scan point στο δέντρο, για να αρχίσει η scan από το μικρότερο value του. Βοηθά στην εύρεση values που είναι μικρότερα από κάποιο συγκεκριμένο value.
- get_value_point: Επιστρέφει scan point στην πρώτη εγγραφή του block στο οποίο ανήκει ή θα ανήκε το value για το οποίο ψάχνουμε. Χρησιμοποιεί για αναζητήσεις μεγαλύτερου, μεγαλύτερου ή ίσου και ίσου.
- scan_get_next_value: Παίρνει το επόμενο scan point χρησιμοποιώντας τα στοιχεία που έχει από το τρέχον scan point και αν δεν έχει άλλο value, γυρνάει null.
- scan_check: Ελέγχει, ανάλογα με τον τελεστή, εάν ισχύει η συνθήκη και τότε επιστρέφει 1 (αν προορίζεται για εκτύπωση), 0 (αν πρόκειται να πάρει το επόμενο value) και -1 (αν πρόκειται να σταματήσει).
- Ascend: Χρησιμοποιείται όταν χρειάζεται να κάνουμε προώθηση προς τα πάνω και να κάνουμε ορισμένες αλλαγές σε blocks ευρετηρίου.

Σχόλια:

Σε όλη την έκταση της άσκησης έχουν προστεθεί επιπλέον σχόλια, προκειμένου να είναι ευνόητη η μέθοδος που ακολουθείται.