

---

## 1<sup>η</sup> Άσκηση-Επεξεργασία Αρχείων

Βαγγέλης Αθανασάκης 2019030118

---

Σκοπός της πρώτης άσκησης ήταν η διερεύνηση του πιο αποδοτικού τρόπου οργάνωσης αρχείου το οποίο θα χρησιμοποιηθεί για αναζητήσεις.

- Αρχικά, δημιουργήθηκε η κλάση «FileManager» η οποία έχει μεθόδους (όπως πχ. ReadBlock, WriteBlock, OpenFile) οι οποίες είναι χρήσιμες για την παρακάτω επεξεργασία των αρχείων.
- Στην συνέχεια δημιουργήθηκε η κλάση «FileGenerator» η οποία μπορεί μέσω της μεθόδου «DataFileGenerator» να φτιάξει ένα αρχείο(DataFile) όπου κάθε σελίδα να έχει 4 Records(κλειδί-πληροφορία) και μέσω της μεθόδου «KeyFileGenerator» να φτιάξει ένα αρχείο(KeyFile) όπου κάθε σελίδα να έχει 16 Records(Κλειδί-θέση κλειδιού στο αρχείο κλειδί-πληροφορία). Το KeyFile περιέχει τα κλειδιά του DataFile και την θέση της σελίδας τους στο DataFile. Τα κλειδιά και η πληροφορία παράγονται απο γεννήτριες τυχαίων ακεραίων-αλφαριθμητικών.
- **Για τον Α τρόπο οργάνωσης αρχείου** το αρχείο που περιέχει τα κλειδί-πληροφορία(DataFile) σαρώνεται σειριακά. Αυτή η λειτουργία υλοποιείται απο την μέθοδο «SerialSearchingData» της κλάσης «Searching» η οποία εκτελεί σειριακή αναζήτηση, φέρνοντας μια σελίδα την φορά στην κεντρική μνήμη, μετατρέποντας τα byte arrays σε node arrays μέσω της «BlockToNodeArrayProccess» της κλάσης «BlockToNodeArray» και αναζητώντας το κλειδί που ζητείται κάθε φορά.
- **Για τον Β τρόπο οργάνωσης αρχείου** δημιουργείται το KeyFile και εκτελείται εκεί ο αλγόριθμος της σειριακής αναζήτησης μέσω της «SerialSearchingKeys» της κλάσης «Searching». Η μέθοδος αυτή φέρνει μία σελίδα κάθε φορά στην κεντρική μνήμη, την μετατρέπει σε Array από 16 IndexNodes, αναζητεί εκεί το κλειδί και τέλος, όταν βρεί το IndexNode με το κλειδί, παίρνει την πληροφορία της θέσης σελίδας του DataFile και αναζητεί σε εκείνη την σελίδα του DataFile το κλειδί.
- **Για τον Γ τρόπο οργάνωσης αρχείου** εκτελείται η μέθοδος «externalSort» της κλάσης «ExternalSort» η οποία διαβάζει όλο το DataFile στην κεντρική μνήμη , μετατρέποντάς το μέσω της «BlockToNodeArrayProccess» σε Array απο Nodes. Στην συνέχεια, εκτελείται αλγόριθμος «QuickNodeSort» ο οποίος ταξινομεί το Array με βάση το κλειδί. Έπειτα, δημιουργείται ένα νέο αρχείο δεδομένων(SortedDataFile), τύπου DataFile, μέσω της μεθόδου

«SortedDataFileGenerator» της κλάσης «FileGenerator» με ταξινομημένα τα κλειδιά. Τέλος, εκτελείται αλγόριθμος αναζήτησης BinarySearch στο ταξινομημένο αρχείο για την εύρεση του κλειδιού μέσω της μεθόδου «BinarySearchingData» της κλάσης «Searching».

- **Για τον Δ τρόπο οργάνωσης αρχείου** εκτελείται η μέθοδος «externalSort» της κλάσης «ExternalSort» η οποία διαβάζει όλο το αρχείο των κλειδιών(KeyFile) στην κεντρική μνήμη και το μετατρέπει μέσω της «BlockToIndexNodeArrayProccess» σε Array από IndexNodes. Στην συνέχεια, εκτελείται ο αλγόριθμος «QuickIndexNodeSort» ο οποίος ταξινομεί το Array με βάση το κλειδί. Έπειτα, δημιουργείται ένα νέο αρχείο κλειδιών(SortedKeyFile), τύπου KeyFile, μέσω της μεθόδου «SortedKeyFileGenerator» της κλάσης «FileGenerator» με ταξινομημένα τα κλειδιά. Σε αυτό το αρχείο εκτελείται αλγόριθμος αναζήτησης BinarySearch μέσω της μεθόδου «BinarySearchingKeys» της κλάσης «Searching», βρίσκεται το IndexNode με το κλειδί και επιστρέφεται η θέση της σελίδας που περιέχει το κλειδί στο αρχικό αρχείο(DataFile). Τέλος, εκτελείται δυαδική αναζήτηση στο block του DataFile για την εύρεση του κλειδιού.

### **Τεκμηρίωση Αποτελεσμάτων**

Μέθοδος	Α. Τρόπος Οργάνωσης	Β. Τρόπος Οργάνωσης	Γ. Τρόπος Οργάνωσης	Δ. Τρόπος Οργάνωσης	Γ.Απόδοση Εξωτερικής ταξινόμησης	Δ.Απόδοση Εξωτερικής ταξινόμησης
Απόδοση(Disk Accesses)	87.703	23.473	16	15	562.498	62.500

- **Σύγκριση μεθόδων Α-Β:** Προκύπτει ότι ο Β τρόπος οργάνωσης αρχείου είναι πολύ πιο αποδοτικός από τον Α καθώς ο Β πραγματοποιεί πολλές λιγότερες προσβάσεις στον δίσκο για την εύρεση του κλειδιού. Αυτό συμβαίνει γιατί, στην ουσία με μία πρόσβαση δίσκου στον Α τρόπο οργάνωσης αρχείου μπορούν να ελεγχθούν 4 κλειδιά (τόσα έχει η κάθε σελίδα αφού χωράει 4 εγγραφές) ενώ με τον Β τρόπο μπορούν να ελεγχθούν 16 κλειδιά (τόσα έχει η κάθε σελίδα στο αρχείο των κλειδιών αφού χωράει 16 IndexNodes). Το αρχείο κλειδιών αποτελείται από λιγότερες σελίδες.
- **Σύγκριση μεθόδων Γ-Δ:** Προκύπτει ότι ο Δ τρόπος οργάνωσης αρχείου είναι πιο αποδοτικός από τον Γ καθώς πραγματοποιεί λιγότερες προσβάσεις στον δίσκο για την εύρεση του κλειδιού. Αυτό συμβαίνει γιατί στον Γ τρόπο οργάνωσης αρχείου η αναζήτηση πραγματοποιείται σε ολόκληρο το αρχείο των εγγραφών ενώ στον Δ μόνο στο αρχείο των κλειδιών. Επίσης και εδώ μία πρόσβαση δίσκου

με τον Γ τρόπο οργάνωσης ελέγχει 4 κλειδιά ενώ με τον Δ τρόπο οργάνωσης ελέγχει 16 (Το αρχείο των κλειδιών αποτελείται από λιγότερες σελίδες).

- **Σύγκριση μεθόδων A-B,Γ-Δ:** Προκύπτει ότι οι δύο τελευταίοι τρόποι οργάνωσης αρχείου είναι πολύ πιο αποδοτικοί από τους δύο πρώτους με την προϋπόθεση ότι το αρχείο είναι ταξινομημένο. Αυτό είναι αναμενόμενο καθώς στους Γ-Δ τρόπους οργάνωσης το αρχείο είναι ταξινομημένο και εφαρμόζεται ο αλγόριθμος δυαδικής αναζήτησης ο οποίος είναι πολύ πιο αποδοτικός από αντίστοιχο αλγόριθμο σειριακής αναζήτησης. Αν το αρχείο δεν είναι ταξινομημένο η απόδοση στους Γ-Δ τρόπους πέφτει δραματικά.
- **Σύγκριση εξωτερικής ταξινόμησης στην Γ και Δ περίπτωση:** Προκύπτει ότι η απόδοση είναι πολύ καλύτερη στην περίπτωση Δ καθώς ταξινομείται το αρχείο των κλειδιών και όχι των δεδομένων. Το αρχείο των κλειδιών είναι 4 φορές μικρότερο σε μέγεθος σελίδων με αποτέλεσμα να χρειάζονται λιγότερες προσβάσεις στον δίσκο για να ταξινομηθεί.
- **Πηγές:** Για την υλοποίηση της άσκησης χρησιμοποιήθηκε υλικό του φροντιστηρίου το οποίο τροποποιήθηκε ελαφρώς (Κώδικας Binary Search ο οποίος τροποποιήθηκε για να διαχειρίζεται Nodes, κώδικες γεννητριών τυχαίων αριθμών και αλφαριθμητικών, κλάση MultiCounter για μετρήσης απόδοσης, Κλάση BlockToNodeArray για μετατροπή byte arrays σε nodes). Παράλληλα, χρησιμοποιήθηκε υλικό από το διαδίκτυο το οποίο επίσης τροποποιήθηκε ελαφρώς (αλγόριθμος ταξινόμησης QuickSort από την διεύθυνση: <https://www.geeksforgeeks.org/quick-sort/>).