

ΑΛΕΞΑΝΔΡΟΣ ΚΟΣΜΑΣ ΑΜ:1115201700299

Το σκεπτικό μου βασίζεται να διαχειριστώ την μνήμη μας με έναν εύκολο τρόπο για εμένα, γι αυτό έφτιξα δύο δικές μου δομές την `HP_info` και τις τοποθέτησα στο `heap_file` που μεσω αυτής και την χρήση της `memset` περνάω την πληροφορία αν έχω αρχείου σωρού στην `create`, και κανω τον αντιστοιχο έλεγχο στην `open` χρησιμοποιώντας μια σειρά απο συναρτήσεις που μας έχουν δοθεί. Η άλλη δομή που έχω χρησιμοποιήσει είναι η `Block` για τον ίδιο λόγο για να έχω ευκολία στην διαχείριση της μνήμής, και την δημιουργήσει έτσι που με τον πίνακα(`records[8]`) που της έχω ορίσει εντός, να μπόρω κάθε φορά με το που μπαίνω στην `insert`, αφού ελέγξω σε ποια θέση είμαι και αν χρειάζομαι νέο `block`, να περνώ τα `data` μου από την `BF_Block_GetData` και να τα περνώ στην δικιά μου μεταβλήτη, χρησιμοποιώντας ταυτόχρονα και την άλλη μεταβλητή της δομής μου την `count` για να κρατάω πότε θα γεμίσει το μπλογκ μου. Όσο έχει χώρο(<7) αυξάνω το `count` μου και εισάγω στην μεταβλητη μου την αντιστοιχη εγγραφή. Όταν γεμίσει δημιουργώ νέο `block` με `allocate` και κάνω την αντίστοιχη δουλεία μεταφοράς δεδομένων με `prin`. Στην `HP_PrintAllEntries` καλώ δυο επαναλήψεις(`for`) με την πρώτη να τρέχει τα `blocks` που έχουμε εισάγει και την άλλη για τις εγγραφές χρησιμοποιω παλι την δομή μου `Block` ώστε να διαχειριστώ τα `data`, χρησιμοποιώντας συναρτήσεις που μας δίνονται και με κάποιους ελέγχους αν είναι `NULL` και αντίστοιχα με την χρήση της `strcmp` με ποια `attrName` ή `value` ταυτίζονται οι εγγραφές μας εκτυπώνουμε τα αντιστοιχα δεδομένα. Για την `HP_GetEntry` δεν καταφερα να την ολοκληρώσω την έχω βάλει στα σχόλια να μην βγάζει `error`, την άφησα μήπως εκτιμηθεί η προσπάθεια.