Ονοματεπώνυμο: Γεώργιος Τσίρης
Αριθμός Μητρώου: 1115201700173

<u>Σχεδιαστικές Επιλογές:</u>

1. Το πρώτο μπλοκ ενός αρχείου σωρού (heap file) διακρίνεται από τον χαρακτήρα 'h' εκεί όπου αρχίζουν τα δεδομένα του. Ο υπόλοιπος χώρος του πρώτου μπλοκ δεν χρησιμοποιείται.
2. Τα υπόλοιπα μπλοκ πλην του πρώτου, αρχίζουν με ένα ακέραιο record_counter, ο οποίος ανα πάσα στιγμή έχει καταμετρημένο το τρέχον πλήθος εγγραφών που διαθέτει το συγκεκριμένο μπλοκ. Από εκεί και πέρα, όλος ο υπόλοιπος χώρος είναι για την αποθήκευση εγγραφών.
3. Η υλοποίηση δεν απαιτεί κάποια επιπρόσθετη δομή και ως εκ'τούτου η συνάρτηση HP_Init() δεν περιλαμβάνει κάποια αρχικοποίηση.

<u>Απάντηση στην ερώτηση Bonus:</u>

Με την βοήθεια της εντολής strace -c ./build/runner παρατηρούμε ότι:

- Πλήθος read με LRU: 106521
- Πλήθος read με MRU: 56922

Παρατηρούμε ότι το πλήθος των read είναι σαφώς μεγαλύτερο στην LRU (περίπου διπλάσιο).
Αυτό οφείλεται στο εξής:
Η συνάρτηση TestFileScan() αποτελεί περίπτωση αλγορίθμου εμφωλιασμένων βρόγχων και άρα έχουμε ότι για κάθε j γίνονται όλες οι επαναλήψεις του εσωτερικού βρόγχου (το id παίρνει όλες τις ακέραιες τιμές από [RECORDS_NUM, 0]):

- Στην στρατηγική αντικατάστασης του Λιγότερο Πρόσφατα Χρησιμοποιημένου (LRU) όταν το id φτάνει κάθε φορά στο 0 τα μπλοκ που είχαν χρησιμοποιηθεί στις αρχικές τιμές του id (δηλ για id = RECORDS_NUM, id = RECORDS_NUM – 1, …) έχουν πλέον αντικατασταθεί όντας τα λιγότερο πρόσφατα. Αυτά ακριβώς τα μπλοκ είναι όμως που θα χρειαστούν μόλις αρχίσει η εκτέλεση του εσωτερικού βρόγχου για το επόμενο j. Αναγκαστικά λοιπόν θα πρέπει να διαβαστούν (read) από τον δίσκο, μιας και δεν είναι πια στην ενδιάμεση μνήμη.
- Αντιθέτως, στην στρατηγική αντικατάστασης του Πλέον Πρόσφατα Χρησιμοποιημένου (LRU) όταν το id φτάνει κάθε φορά στο 0 τα μπλοκ που είχαν χρησιμοποιηθεί στις αρχικές τιμές του id (δηλ για id = RECORDS_NUM, id = RECORDS_NUM – 1, …) βρίσκονται ακόμα στην ενδιάμεση μνήμη όντας τα λιγότερο πρόσφατα. Αφού αυτά είναι που θα χρειαστούν για να εκτελεστεί ο εσωτερικός βρόγχος για το επόμενο j, γλυτώνουμε το διάβασμα από τον δίσκο για τα συγκεκριμένα μπλοκ. Το πόσες λιγότερες read θα χρειαστούμε είναι ανάλογο του μεγέθους της ενδιάμεσης μνήμης (buffer).

Σημείωση: Αν στο buffer χωρούσαν αρκετά μπλοκ για όλες τις εγγραφές οι δύο παραπάνω στρατηγικές θα είχαν ακριβώς το ίδιο κόστος εισόδου/εξόδου (όπου το κόστος εισόδου/εξόδου αυξάνεται κατά 1 ανά μεταφορά μπλοκ).