

Παράλληλος Προγραμματισμός 2018
Προγραμματιστική Εργασία #1
Ονοματεπώνυμο: Ζουγγουρίδης Παναγιώτης Σωτήριος
ΑΜ: Π2014174

Αρχικά στον κώδικα μου ορίζω τον πίνακα που ζητάει η άσκηση και κάνω αρχικοποίηση βάζοντας παντού την τιμή 1.0.

Στη συνέχεια ξεκινάει η διαδικασία GETWALLTIME η οποία αρχίζει να μετράει τον χρόνο που θα χρειαστεί μέχρι να προσπελαστεί όλος ο πίνακας και βγάζει το ts που θα το χρειαστώ αργότερα και μετά ξεκινάει η διαδικασία της προσπέλασης όλου του πίνακα (στο αρχείο matrix1.c ανά στήλη και στο αρχείο matrix2.c ανά γραμμή).

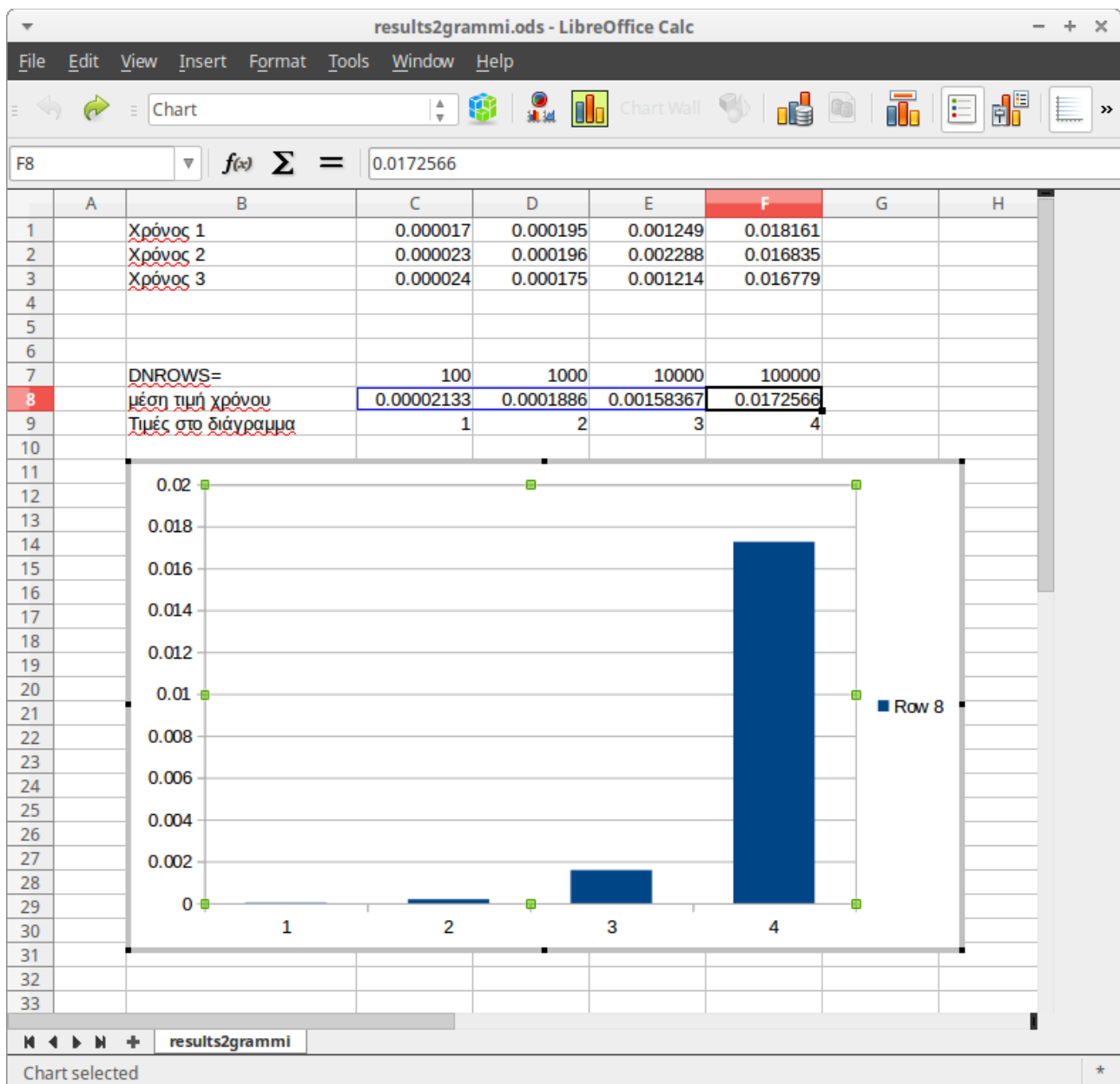
Στη συνέχεια τελειώνει η getwalltime βγάζοντας σαν αποτέλεσμα το te(χρόνος που χρειάστηκε για την προσπέλαση του πίνακα).

Μετά πρέπει να ελέγξω αν είναι σωστά τα αποτελέσματα που θα βγούν με βάση την πράξη που έκανα πριν (στην συγκεκριμένη περίπτωση
table[j*NCOLS +i]=table[j*NCOLS +i]+13;)

Αν είναι σωστό το αποτέλεσμα δηλαδή ίσο με 14 τότε παίρνει το te και το ts από την getwalltime και τα αφαιρεί για να υπολογίσει το χρόνο που χρειάστηκε για να γίνει η προσπέλαση του πίνακα.

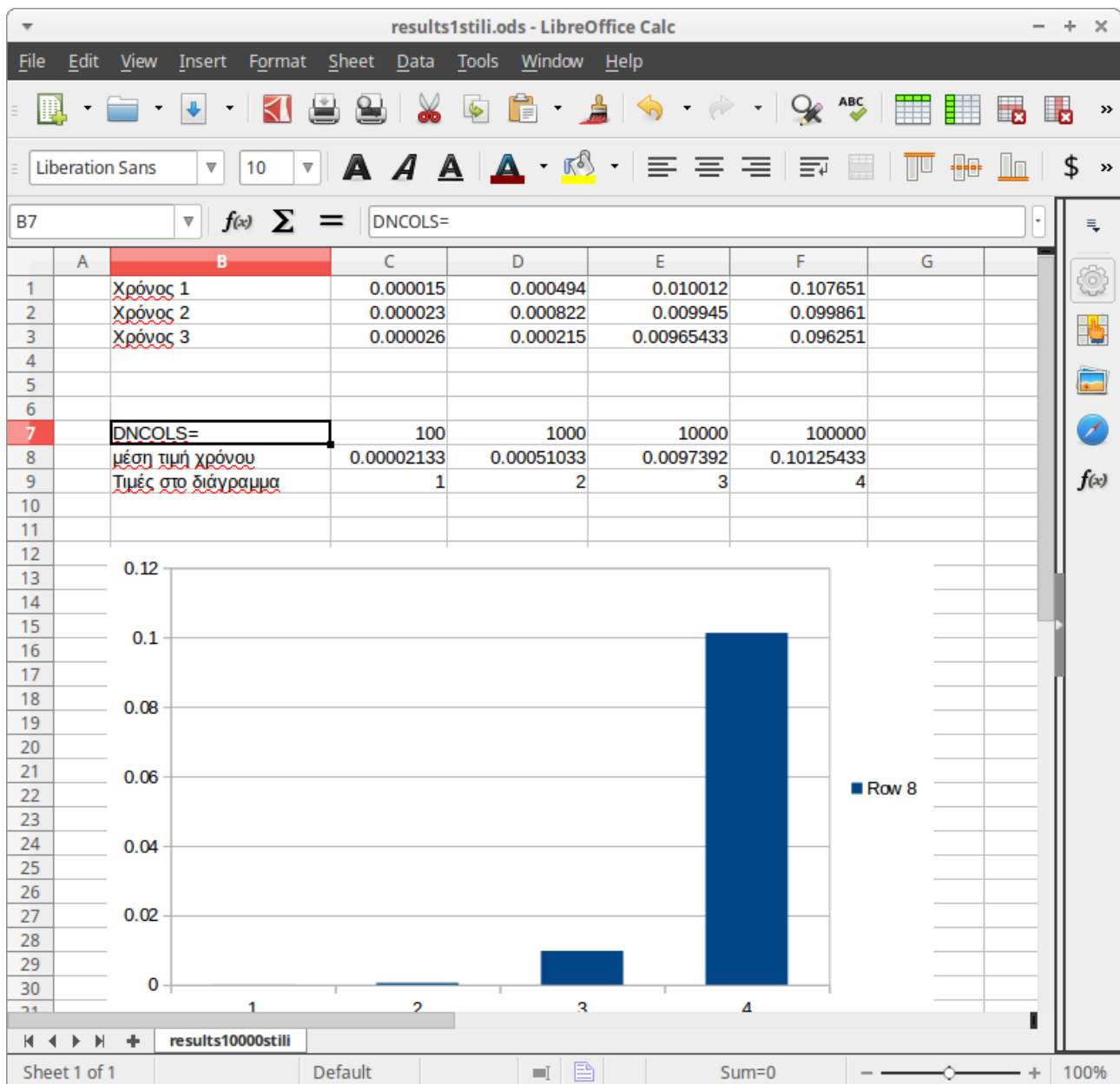
Τέλος αδειάζω τη μνήμη του πίνακα.

Παρακάτω έχω τα διαγράμματα και τη διαδικασία δημιουργίας τους.



Η διαδικασία για τις γραμμές έχει ως εξής

1. τρέχω 3 φορές το αρχείο c για 100 για 1000 για 10000 και για 100000
οπότε βγάζω 3 αποτελέσματα για τον κάθε αριθμό γραμμών.
2. κρατάω τα αποτελέσματα ως χρόνος 1 χρόνος 2 και χρόνος 3
3. βρίσκω για κάθε περίπτωση την μέση τιμή του χρόνου που χρειάστηκε
4. εισάγω τις μέσες τιμές σε ένα διάγραμμα



Η διαδικασία για τις στήλες έχει ως εξής

1. τρέχω 3 φορές το αρχείο c για 100 για 1000 για 10000 και για 100000
οπότε βγάζω 3 αποτελέσματα για τον κάθε αριθμό στηλών.
2. κρατάω τα αποτελέσματα ως χρόνος 1 χρόνος 2 και χρόνος 3
3. βρίσκω για κάθε περίπτωση την μέση τιμή του χρόνου που χρειάστηκε
4. εισάγω τις μέσες τιμές σε ένα διάγραμμα

Άρα το συμπέρασμα είναι ότι η προσπέλαση των γραμμών είναι πιο γρήγορη από την προσπέλαση των στηλών επειδή τα δεδομένα μεταφέρονται πιο γρήγορα επειδή μεταφέρεται όλη η γραμμή στην κρυφή μνήμη που κάνει τη διαδικασία πιο γρήγορη.