ΣΕΙΡΑ ΕΡΓΑΣΙΩΝ 1

ΒΑΣΙΛΕΙΟΣ ΚΥΡΙΤΣΗΣ 2999

ΑΛΕΞΑΝΔΡΟΣ ΣΤΑΜΟΥΛΟΣ 2954

1.1 FIFO PIPES

Η υλοποίηση έγινε με την βοήθεια ενός struct που περιέχει τα εξής πεδία:

char *buff: ο πίνακας που αποθηκεύονται τα στοιχεία μεταφοράς

int start: η θέση του πίνακα από την οποία διαβάζει το thread

int end: η θέση του πίνακα στην την οποία γράφει το thread

int size: το μέγεθος το buffer

int count: αριθμός των στοιχείων του buffer

int openWrite: μεταβλητή που δείχνει αν ο αγωγός είναι ανοιχτός η όχι για γράψιμο

int openRead: μεταβλητή που δείχνει αν ο αγωγός είναι ανοιχτός η όχι για διάβασμα

int writestart: μεταβλητή που δείχνει αν έχει ξεκινήσει το γράψιμο στον αγωγό η όχι

1.1 FIFO PIPES

Το πρόγραμμα μας χρησιμοποιεί δυο threads για την μεταφορά δεδομένων μέσω ενός ring buffer. Τα δύο threads στο κρίσιμο σημείο τους χρησιμοποιούν τις συναρτήσεις pipe_write, pipre_read και pipe_writeDone κατά βάση και για να αποφύγουμε τις συνθήκες ανταγωνισμού κάνουμε χρήση του αλγορίθμου Peterson.

1.2 ΑΝΑΓΝΏΡΙΣΗ ΠΡΏΤΩΝ ΑΡΙΘΜΏΝ

Η γενική μας ιδέα ήταν να έχουμε έναν global πίνακα από struct, με μέγεθος ίσο με τον αριθμό των threads.

Το struct περιέχει τα εξής πεδία:

int number : ο αριθμός τον οποίο ελέγχουμε

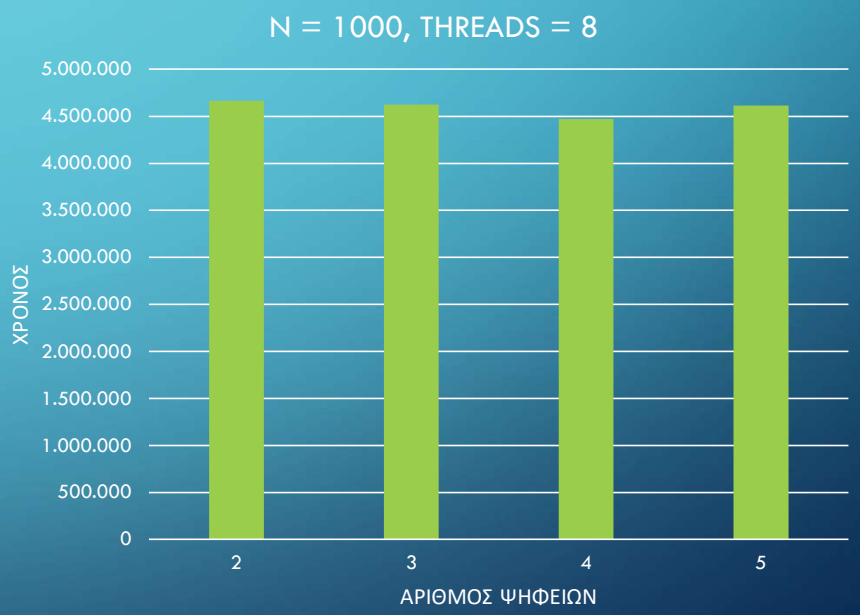
int available : σήμα για το αν το thread είναι διαθέσιμο

int t_terminate : σήμα για το αν το thread τερμάτισε

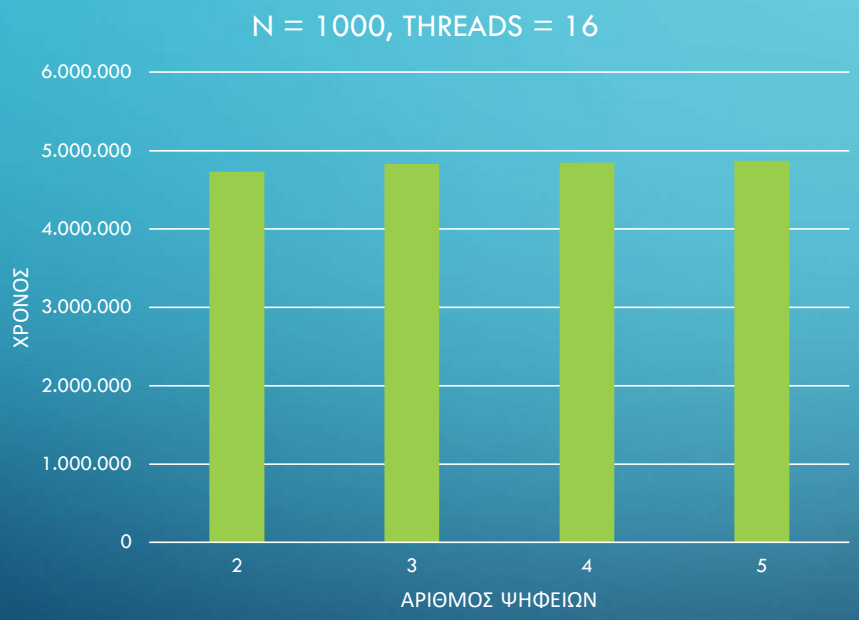
int create : σήμα για το αν το thread έχει δημιουργηθεί

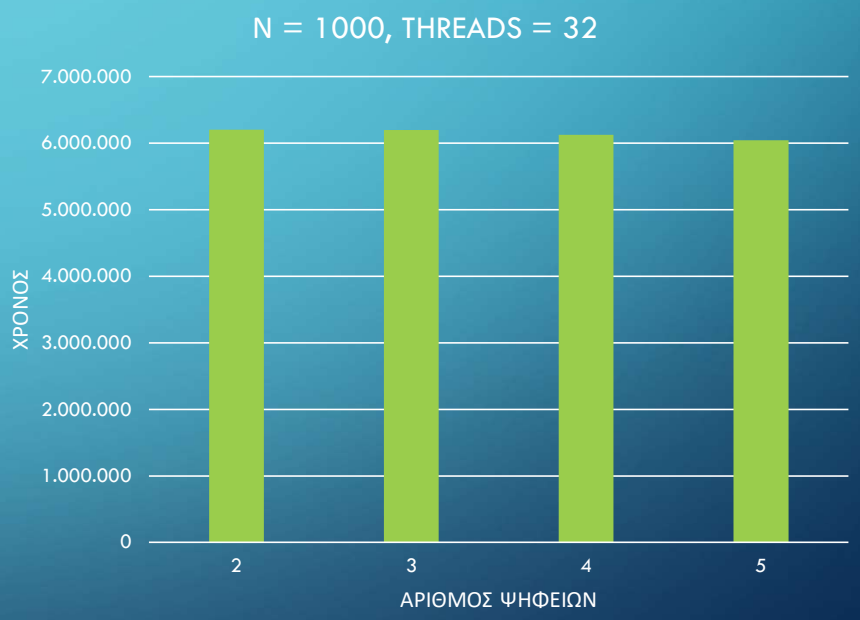
Επιπλέον έχουμε το global σήμα terminate, το οποίο ενημερώνει τα threads να τερματίσουν όταν δεν υπάρχουν άλλοι αριθμοί για έλεγχο.

Ο αριθμός των ψηφίων δεν επηρεάζει τον χρόνο εκτέλεσης αισθητά, ενώ ο αριθμός των threads στην αρχή μειώνει τον χρόνο, ενώ όταν τα threads φτάσουν μεγάλους αριθμούς μας καθυστερεί η δημιουργία τους



Όντως παρατηρούμε ότι μετά στα 32 threads ο χρόνος εκτέλεσης ανεβαίνει αισθητά





1.3 EXTERNAL MERGESORT

Τρόπος σκέψης:

Αρχικά σχεδιάσαμε το πρόγραμμα μας χωρίς threads καλώντας αναδρομικά μια συνάρτηση , η οποία ήταν τύπου void* και έπαιρνε όρισμα ένα struct έτσι ώστε να γίνει ευκολά η μετάβαση σε πρόγραμμα με threads. Αφουλ κάναμε την υλοποίηση ,καλούμε την συνάρτηση μέσω της pthreadcreate . Τέλος προσθέσαμε κάποια while για την ενεργή αναμονή τερματισμού των threads.

ΨΕΥΔΟΚΩΔΙΚΑΣ

```
Thread_mergesort()
    If(file_size < 64)
       κάνουμε mergesort σε βοηθητικό πίνακα
       return
    Χωρίζουμε το αρχείο στα δυο
    δημιουργούμε δυο threads , ένα για κάθε αρχείο
    περιμένω να τερματίσουν τα threads
    ενώνουμε τα δυο ταξινομημένα αρχεία σε ένα ταξινομημένο
    return
```