

Γιαννούτσος Ανδρέας 1115201700021

Λειτουργικά Συστήματα Εργασία 1

Σχετικά με τη μεταγλώττιση και ο τρέξιμο

Το πρόγραμμα μεταγλωττίζεται με την εντολή **make mainexe** , η οποία δημιουργεί το εκτελέσιμο **mainexe** το εκτελέσιμο τρέχει με την παρακάτω εντολή:

./mainexe arg1 arg2 arg3

arg1 είναι ο αριθμός των peers

arg2 είναι το ποσοστό readers και writers με τιμές από [1 έως 100] που 1 σημαίνει 1% readers και 99% και για παράδειγμα το 30 σημαίνει 30% readers και 70% writers

arg3 είναι το μέγεθος της διαμοιραζόμενης μνήμης

Ένα ενδεικτικό τρέξιμο θα μπορούσε να είναι ./mainexe 100 70 100 καθώς οι συντελεστές στο wait είναι τέτοιο ώστε να τρέχει με τέτοιο αριθμό μνήμης και peers καλύτερα.

Σχετικά με την υλοποίηση

Στο πρόγραμμα έχουν χρησιμοποιηθεί κομμάτια κώδικα από το εργαστήριο 2 και από το βιβλίο για το πρόβλημα readers writers.

Υπάρχουν 3 παραπάνω αρχεία .C τα semaphores.c shmemory.c tester.c.

Το πρώτο έχει συναρτήσεις για την αρχικοποίηση και την αλλαγή των σημαφόρων. Το 2^ο έχει και αυτό συναρτήσεις για τον ορισμό και διαγραφή της κοινής μνήμης. Το 3^ο περιέχει 2 επαναλήψεις που δοκιμάζουν το πρόγραμμα για διαφορετικές τιμές μνήμης και peers.

Χρησιμοποιούνται 3 ομάδες σημαφόρων memory size μεγέθους ο καθένας και ένας δυαδικός. Οι writeld readld counterld printld. Ο 1^{ος} ελέγχει τους writers, ο 2^{ος} τους readers, ο 3^{ος} ελέγχει τον μετρητή για το πόσες διεργασίες διαβάζουν τον ίδιο πόρο κάθε φορά, ώστε να γνωρίζουμε αν θα αποκλείσουμε τους εγγραφείς και ο 4^{ος} χρησιμοποιείται ώστε οι διεργασίες να εκτυπώνουν τα αποτελέσματα σε ένα αρχείο όλες μαζί.

Η διαμοιραζόμενη μνήμη έχει τον πίνακα δοσμένου μεγέθους entries από μία δομή που περιέχει μετρητή αναγνώσεων και εγγραφών, καθώς επίσης και έναν πίνακα για τους μετρητές αναγνωστών σε κάθε entry. Ακόμα υπάρχει και ένας πίνακας 3 στοιχείων όπου καταγράφεται ο

συνολικός χρόνος αναμονής και προστίθεται από κάθε διεργασία για τα στατιστικά τελικά αποτελέσματα.

Ο αλγόριθμος για την υλοποίηση των readers και writers είναι αυτός που έχουμε διδαχθεί στο μάθημα με τους 3 σημαφόρους τον μετρητή των αναγνώσεων και τους ελέγχους του.

Ο χρόνος αναμονής υπολογίζεται σε κάθε διεργασία από την στιγμή που θα κάνει αίτηση για κάποιον σημαφόρο μέχρι το σημείο που θα μπει μέσα στο critical section. Ο χρόνος υπολογίζεται σε μικρο-δευτερόλεπτα. Στο critical section υπάρχει ξεχωριστή καθυστέρηση με τον εκθετικό χρόνο όπως είχε αναφερθεί το μάθημα με τον τύπο $\ln(U)/\lambda$ με $U[0,1]$ και λ μια σταθερά που ορίζεται με `#define lambda` στην αρχή του `main.c`

Τα στατιστικά για τους χρόνους αναμονής αλλά και η επαλήθευση των αποτελεσμάτων γίνεται σε 2 αρχεία, τα `logfile.txt` και `logstat.txt`. Στο `logfile` καταγράφεται στην κάθε εκτέλεση του προγράμματος ο μέσος χρόνος και ο ολικός χρόνος για την ανάγνωση και την εγγραφή για κάθε διεργασία. Στο τέλος του γίνεται επαλήθευση των αποτελεσμάτων καθώς αθροίζονται οι μετρητές καταγραφών και αναγνωστών από την διαμοιραζόμενη μνήμη και γίνεται η σύγκριση με το `peers*LOOPS` που είναι το πόσες φορές έγινε ένα read or write. Αυτοί οι δύο αριθμοί είναι πάντα ίσοι. Στο `logstat` καταγράφεται σε κάθε εκτέλεση και δεν διαγράφεται με την σειρά το άθροισμα των μέσων χρόνων αναμονής, ο μέσος χρόνος αναμονής ανάγνωσης και ο μέσος χρόνος αναμονής εγγραφής. Τα αποτελέσματα είναι σε λίστα και διευκολύνεται η χρήση τους σε πίνακες και γραφήματα στο `εξελ`.

Σχετικά με τα αποτελέσματα

Το πρόγραμμα `tester` τρέχει το κύριο πρόγραμμα με διαφορετικές τιμές μνήμης και `peers` για συγκεκριμένο ποσοστό `R/W`. Πιο αναλυτικά έχει 2 for:

For μέγεθος μνήμης από 0 μέχρι 100

For αριθμός `peers` από 100 μέχρι 0

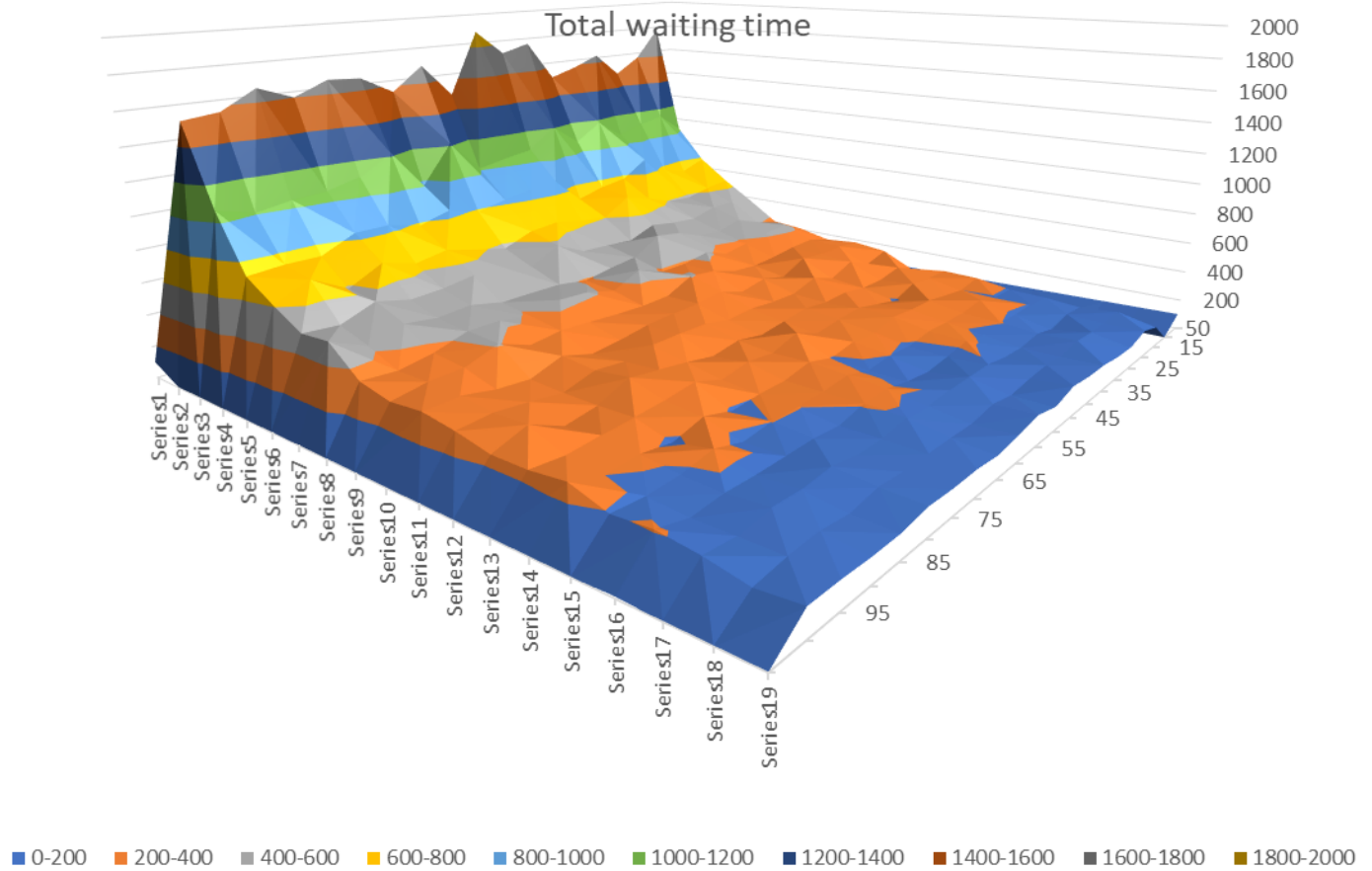
`exemain peers 50` μνήμη

τα αποτελέσματα για διαφορετικά ποσοστά βρίσκονται σε αρχεία `εξελ` μαζί και με τις γραφικές παραστάσεις τους.

Ακολουθούν γραφικές παραστάσεις και σχολιασμός των δεδομένων.

Οι γραφικές παραστάσεις έχουν γίνει με ποσοστό `R/W = 50%`.

Γράφημα αποτελεσμάτων



Στον άξονα Z είναι ο χρόνος και στους άξονες X και Y είναι το μέγεθος της μνήμης και ο αριθμός των peers.

Στο παρακάτω γράφημα όμως αποτυπώνονται καλύτερα τα αποτελέσματα.

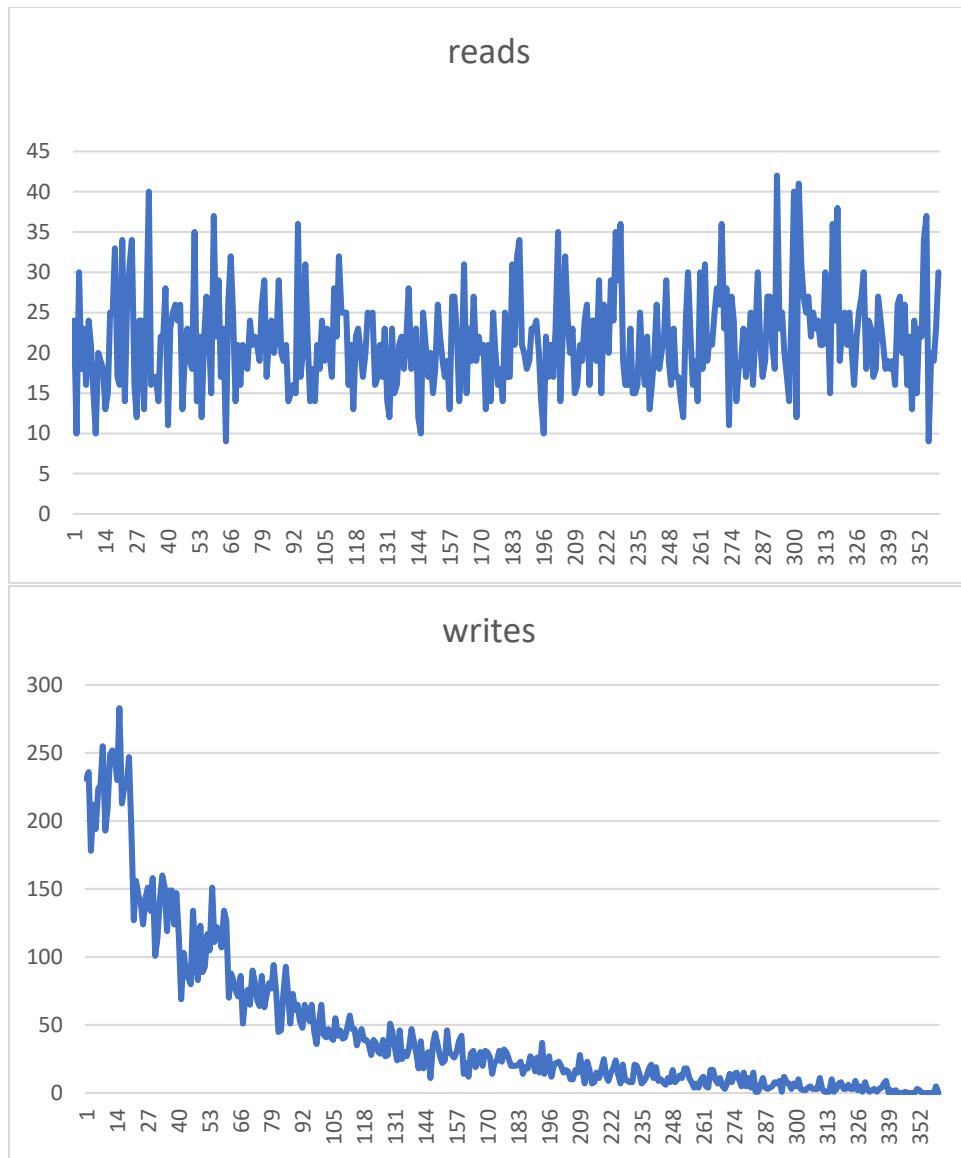


Σε αυτό το γράφημα ο άξονας X είναι το μέγεθος της μνήμης αλλά σε αυτόν τον αλλάζει και το αριθμός των peers , δηλαδή είναι η προβολή του πάνω γραφήματος ανά περίοδο τον αριθμό των peers.

Και στα δύο παρατηρούμε πως τον μεγαλύτερο ρόλο για τον χρόνο αναμονής παίζει το μέγεθος της μνήμης. Καθώς όσο μεγαλύτερο είναι τόσο μικρότερος είναι ο χρόνος αναμονής.

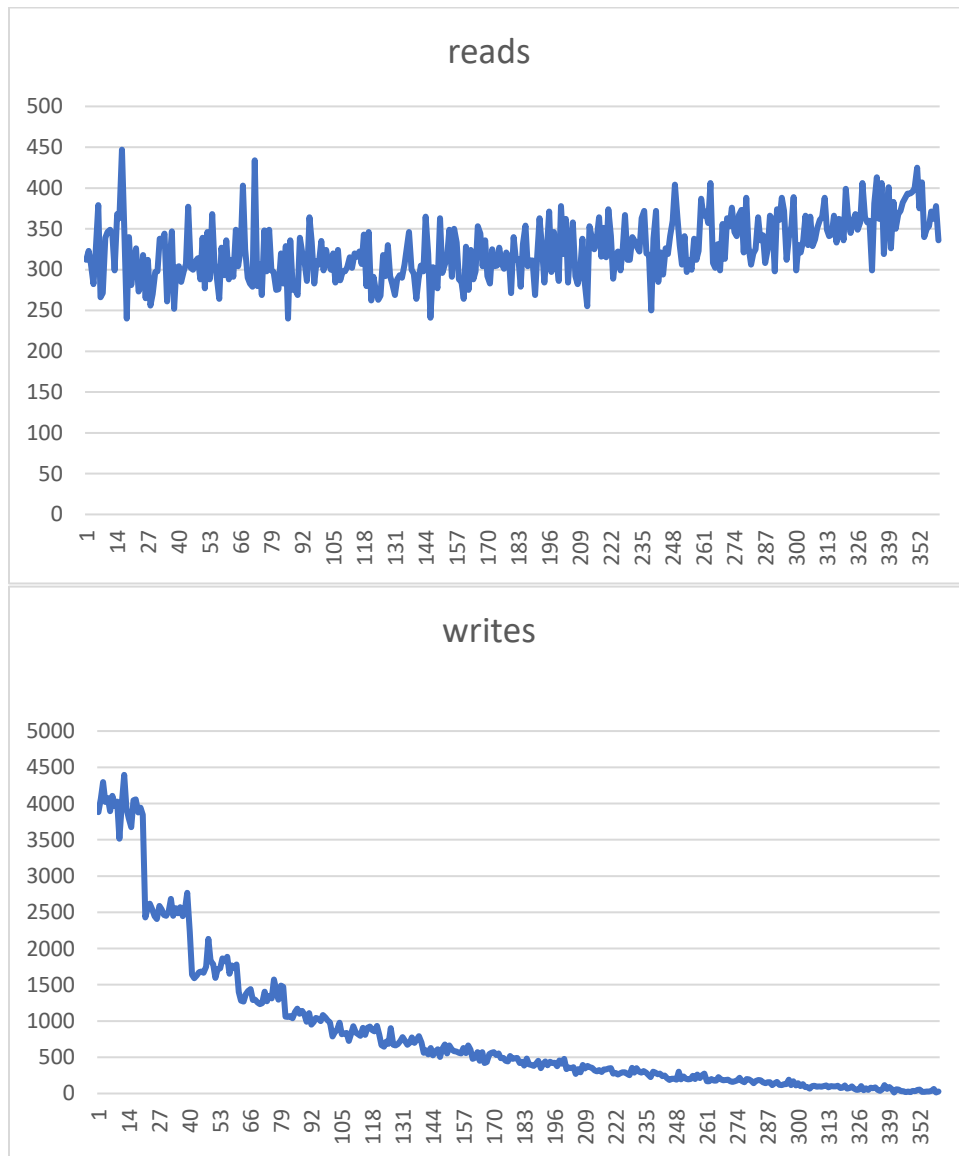
Ο αριθμός των peers όμως παίζει και αυτός ρόλο καθώς όταν η διαφορά μνήμης και Peers είναι μεγάλη τότε ο χρόνος αναμονής αυξάνεται η μειώνεται αρκετές τάξεις μεγέθους περισσότερο από την διαφορά τους.

Τέλος σχετικά με τον χρόνο αναμονής read και write έχουμε τις παρακάτω γραφικές παραστάσεις για ποσοστό 20%



Παρατηρούμε πως οι εγγραφείς επηρεάζονται περισσότερο από την μνήμη καθώς και το ποσοστό αφού όταν μεγαλώσει η μνήμη είναι αμελητέα η καθυστέρηση των πολύ λιγότερων readers.

Και για ποσοστό 80%



Τέλος βλέπουμε πως ο συνολικός χρόνος είναι μεγαλύτερος και για τους δυο αφού οι readers καταλαμβάνουν πιο πολύ τον πόρο τους και επίσης αυτός ο ανταγωνισμός φαίνεται στην αύξηση του χρόνου.