Λειτουργικά Συστήματα(K22): Πρώτη Εργασία

Δημήτρης Φωτεινός

AM:1115201700181

Επεξήγηση κώδικα και παραδοχές:

Αρχικά,περί υλοποίσης του προγράμματος,η ¨δομή' που έχει η διαμοιραζόμενη μνήμη είναι η εξής:(Εικόνα 1.1)

Ακόμη,έχουμε φτιάξει ¨ειδικές' συναρτήσεις για τους σεμαφόρους και την διαμοιραζόμενη μνήμη και κατέπέκταση έχουμε συνολικά 3 πηγαία αρχεία:

main.c Semaphores.c SharedMemory.c

Και συνολικά 2 αρχεία κεφαλίδας:

SharedMemory.h Semaphores.h

Να σημειωθεί ακόμη ότι η βιβλιοθήκη των σεμαφόρων που έχουμε χρησιμοποιήσει είναι η IPC.

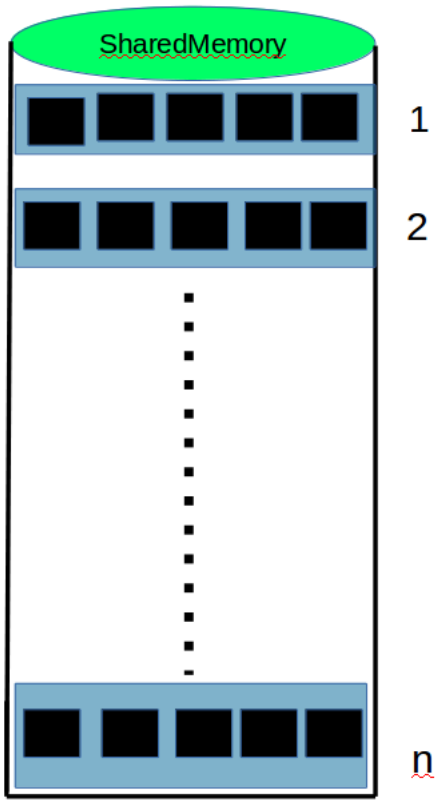
Τα ορίσματα που παίρνει η main(Coordinator) είναι:

π.χ: ./exe 10 5 0.5

Όπου **exe** το όνομα του εκτελέσιμου

- Το πρώτο όρισμα που δίνουμε αφορά τα entries που θέλουμε να έχει η διαμοιραζόμενη μνήμη, στο συγκεκριμένο παράδειγμα τα **Entries = 10**.
- Το δεύτερο όρισμα που δίνουμε αφορά τα Peers(διεργασίες) που θέλουμε να παράγει ο Coordinator(main), στο συγκεκριμένο παράδειγμα **Peers = 2**.
- Το τρίτο όρισμα που δίνουμε αφορά την αναλογία των Readers/Writers, δηλαδή πόσους Readers και πόσους Writers θέλουμε να παράγει η κάθε διεργασία, στο συγκεκριμένο παράδειγμα **RatioRW = 0.5**.

Εικόνα 1.1:



Όπου κάθε μπλε ορθογώνιο πλέγμα,αντιπροσωπεύει την δομή του Entry που έχουμε, δηλαδή:


```
struct Entry{
    Int WritersCounter;
    Int ReadersCounter;
    Int SharedReaders;
    Int SemaphoreWrt;
    Int SemaphoreMutex;
}
```

Να σημειωθεί ακόμη ότι οι επαναλήψεις για τις οποίες "ενεργοποιούνται" οι διεργασίες έχουν οριστεί με **#define ITER 10** στο πρόγραμμα.

Ο εκθετικός χρόνος για τον οποίο "εκτελεί" το critical section της μια διεργασία έχει οριστεί απ'τον λόγο:

$$\mathbf{T = MULT \cdot \frac{-ln(U)}{\lambda}}$$

Όπου αναφορικά έχουμε:

- MULT=Ένας αριθμός όπου πολλαπλασιάζουμε αυτό τον λόγο για να έχουμε "ρεαλιστικά" νούμερα,δηλαδή 3-4 δευτερόλεπτα κατά μέσο όρο για κάθε διεργασία.Στο πρόγραμμα αυτός ο αριθμός έχει οριστεί με #define MULT 1500
- U=Ένας "τυχαίος" αριθμός από 0 έως 1.
- λ=Ένας αριθμός τον οποίο διαιρούμε τον λογάριθμο,στο πρόγραμμα αυτό ο αριθμός αυτός έχει οριστεί απ'την συνάρτηση: clock().

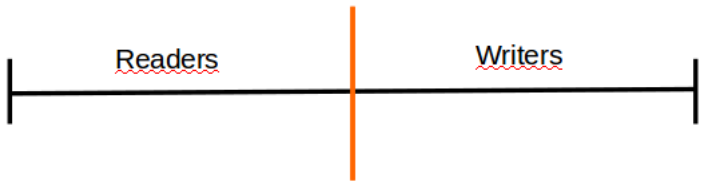
Γενικά, ο κώδικας των Readers και Writers πάρθηκε απ'το βιβλίο(Επιστημονική επιμέλεια ΣΤ.Χατζηευθυμιάδης) στην σελίδα 221.

Τώρα,για την αναλογία των Readers και Writers, έχω υλοποιήσει μια συνάρτηση (την **RWFormat**), η οποία δέχεται ως όρισμα 2 δείκτες σε int,την αναλογία RW και τις επαναλήψεις που θα εκτελέσουν η κάθε μια διεργασία ξεχωριστά. Η συγκεκριμένη συνάρτηση επιστρέφει των αριθμό των Readers και τον αριθμό των Writers που θα έχουμε για **κάθε** διεργασία.

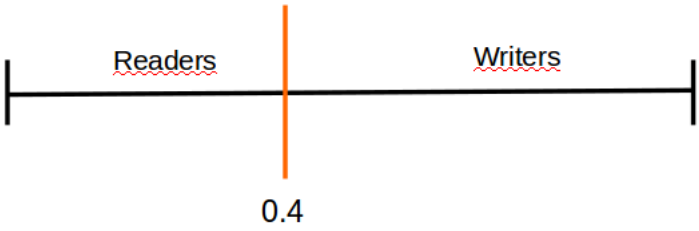
Έτσι,**για παράδειγμα**, αν έχουμε 10 επαναλήψεις και ratio=0.5 , τότε για **κάθε** διεργασία, θα έχουμε ακριβώς 5 readers και 5 writers (με τυχαία σειρά "γέννησης" ,δηλαδή προσπαθούμε όσο το δυνατόν να είναι "μπερδεμένοι" μεταξύ τους και να έχουμε όσο λιγότερο συνεχόμενους readers ή writers).

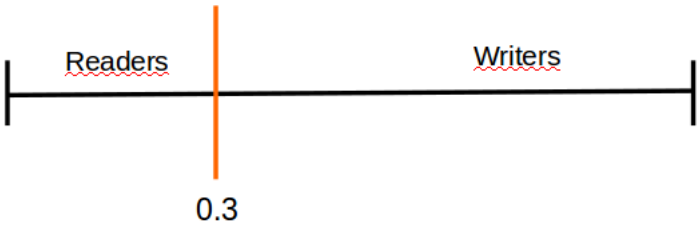
Ας δούμε τώρα, πώς επιλέγουμε "τυχαία" κάθε φορά η διεργασία (σε κάθε επανάληψη δηλαδή) να παράγει είτε έναν reader είτε έναν writer:

Αν φανταστούμε πως έχουμε ένα σημείο αναφοράς,όπως φαίνεται και στο παρακάτω σχήμα(συγκεκριμένα το ratio που έχουμε πάρει ως όρισμα απ'την γραμμή εντολών):



Εικόνα 1.2:





Στην εικόνα αυτό που συμβαίνει είναι:

Αρχικά, επιλέγουμε με την συνάρτηση rand() έναν αριθμό μεταξύ του 0 και του 1. Αν πέσουμε πάνω απ'το (πορτοκαλί) σημείο αναφοράς μας,τότε η

διεργασία μας θα δράσει ως Writer,αλλιώς θα δράσει ως Reader.

Στην αρχή επιλέχθηκε τυχαία ότι η διεργασία θα δράσει ως Reader,οπότε αυτό που κάνουμε κάθε φορά είναι να "μειώνουμε" ουσιαστικά το σημείο αναφοράς μας άμα είναι Reader ή να τον αυξάνουμε αν είναι Writer.

Έτσι,θα είναι πιο "τυχαία" η επιλογή είτε ενός Reader είτε ενός Writer, καθώς σύμφωνα και με τον νόμο των μεγάλων αριθμών **https://el.wikipedia.org/wiki/** κάθε φορά που συμβαίνει το ένα από 2 ενδεχόμενα,τότε την επόμενη φορά είναι "πιο πιθανό" να εμφανιστεί αυτό που δεν εμφανίστηκε στην αρχή.

Τέλος,ας δούμε μερικά παραδείγματα με διαφορετικά ορίσματα στην main για το πρόγραμμα:(Όλα τα αποτελέσματα παρουσιάζονται στο τέλος στο αρχείο: Results.txt

- Για: **./main 10 — 5 — 0.5**:

  Process id: 17629: Average Time: 0.000066 Readers: 5 Writers: 5

  Process id: 17630: Average Time: 0.000048 Readers: 5 Writers: 5

  Process id: 17628: Average Time: 0.000043 Readers: 5 Writers: 5

  Process id: 17627: Average Time: 0.000055 Readers: 5 Writers: 5

  Process id: 17626: Average Time: 0.000069 Readers: 5 Writers: 5

  For entry number: 0
  Total Reads: 0 Total Writes:1

  For entry number: 1
  Total Reads: 1 Total Writes:1

  For entry number: 2
  Total Reads: 5 Total Writes:3

  For entry number: 3
  Total Reads: 6 Total Writes:2

  For entry number: 4
  Total Reads: 3 Total Writes:7

  For entry number: 5
  Total Reads: 4 Total Writes:0

  For entry number: 6
  Total Reads: 1 Total Writes:1

For entry number: 7
Total Reads: 0 Total Writes:0

For entry number: 8
Total Reads: 0 Total Writes:5

For entry number: 9
Total Reads: 5 Total Writes:5

- Για **./main 20 50 0.3:**

  Process id: 17958: Average Time: 0.000012 Readers: 3 Writers: 7

  Process id: 17971: Average Time: 0.000018 Readers: 3 Writers: 7

  Process id: 18002: Average Time: 0.000012 Readers: 3 Writers: 7

  Process id: 18000: Average Time: 0.000019 Readers: 3 Writers: 7

  Process id: 17968: Average Time: 0.000023 Readers: 3 Writers: 7

  Process id: 17981: Average Time: 0.000015 Readers: 3 Writers: 7

  Process id: 17998: Average Time: 0.000014 Readers: 3 Writers: 7

  Process id: 17999: Average Time: 0.000018 Readers: 3 Writers: 7

  Process id: 17985: Average Time: 0.000014 Readers: 3 Writers: 7

  Process id: 17972: Average Time: 0.000021 Readers: 3 Writers: 7

  Process id: 18005: Average Time: 0.000021 Readers: 3 Writers: 7

  Process id: 17977: Average Time: 0.000015 Readers: 3 Writers: 7

  Process id: 17976: Average Time: 0.000015 Readers: 3 Writers: 7

  Process id: 17978: Average Time: 0.000021 Readers: 3 Writers: 7

  Process id: 17980: Average Time: 0.000017 Readers: 3 Writers: 7

  Process id: 18007: Average Time: 0.000014 Readers: 3 Writers: 7

  Process id: 17997: Average Time: 0.000021 Readers: 3 Writers: 7

  Process id: 17966: Average Time: 0.000015 Readers: 3 Writers: 7

  Process id: 17965: Average Time: 0.000020 Readers: 3 Writers: 7

  Process id: 17967: Average Time: 0.000016 Readers: 3 Writers: 7

  Process id: 18006: Average Time: 0.000015 Readers: 3 Writers: 7

  Process id: 17970: Average Time: 0.000019 Readers: 3 Writers: 7

  Process id: 17994: Average Time: 0.000015 Readers: 3 Writers: 7

Process id: 17969: Average Time: 0.000016 Readers: 3 Writers: 7

Process id: 17990: Average Time: 0.000013 Readers: 3 Writers: 7

Process id: 17991: Average Time: 0.000015 Readers: 3 Writers: 7

Process id: 18003: Average Time: 0.000017 Readers: 3 Writers: 7

Process id: 17984: Average Time: 0.000019 Readers: 3 Writers: 7

Process id: 17959: Average Time: 0.000025 Readers: 3 Writers: 7

Process id: 17989: Average Time: 0.000026 Readers: 3 Writers: 7

Process id: 17996: Average Time: 0.000016 Readers: 3 Writers: 7

Process id: 17982: Average Time: 0.000024 Readers: 3 Writers: 7

Process id: 17983: Average Time: 0.000017 Readers: 3 Writers: 7

Process id: 17993: Average Time: 0.000023 Readers: 3 Writers: 7

Process id: 18004: Average Time: 0.000016 Readers: 3 Writers: 7

Process id: 17992: Average Time: 0.000024 Readers: 3 Writers: 7

Process id: 17979: Average Time: 0.000021 Readers: 3 Writers: 7

Process id: 17986: Average Time: 0.000023 Readers: 3 Writers: 7

Process id: 18001: Average Time: 0.000022 Readers: 3 Writers: 7

Process id: 17960: Average Time: 0.000019 Readers: 3 Writers: 7

Process id: 17974: Average Time: 0.000021 Readers: 3 Writers: 7

Process id: 17995: Average Time: 0.000023 Readers: 3 Writers: 7

Process id: 17961: Average Time: 0.000029 Readers: 3 Writers: 7

Process id: 17975: Average Time: 0.000013 Readers: 3 Writers: 7

Process id: 17964: Average Time: 0.000022 Readers: 3 Writers: 7

Process id: 17962: Average Time: 0.000023 Readers: 3 Writers: 7

Process id: 17963: Average Time: 0.000026 Readers: 3 Writers: 7

Process id: 17973: Average Time: 0.000021 Readers: 3 Writers: 7

Process id: 17988: Average Time: 0.000021 Readers: 3 Writers: 7

Process id: 17987: Average Time: 0.000018 Readers: 3 Writers: 7

For entry number: 0
Total Reads: 2 Total Writes:13

For entry number: 1
Total Reads: 11 Total Writes:18

For entry number: 2
Total Reads: 0 Total Writes:20

For entry number: 3
Total Reads: 10 Total Writes:10

For entry number: 4
Total Reads: 10 Total Writes:1

For entry number: 5
Total Reads: 10 Total Writes:40

For entry number: 6
Total Reads: 0 Total Writes:0

For entry number: 7
Total Reads: 14 Total Writes:20

For entry number: 8
Total Reads: 18 Total Writes:25

For entry number: 9
Total Reads: 4 Total Writes:34

For entry number: 10
Total Reads: 15 Total Writes:30

For entry number: 11
Total Reads: 0 Total Writes:10

For entry number: 12
Total Reads: 8 Total Writes:21

For entry number: 13
Total Reads: 2 Total Writes:0

For entry number: 14
Total Reads: 3 Total Writes:22

For entry number: 15
Total Reads: 8 Total Writes:12

For entry number: 16
Total Reads: 14 Total Writes:26

For entry number: 17
Total Reads: 1 Total Writes:4

For entry number: 18
Total Reads: 10 Total Writes:24

For entry number: 19
Total Reads: 10 Total Writes:20

- Για **./main 2 10 0.6**:

  Process id: 18286: Average Time: 0.000183 Readers: 6 Writers: 4

  Process id: 18288: Average Time: 0.000154 Readers: 6 Writers: 4

  Process id: 18295: Average Time: 0.000158 Readers: 6 Writers: 4

  Process id: 18292: Average Time: 0.000266 Readers: 6 Writers: 4

  Process id: 18290: Average Time: 0.000289 Readers: 6 Writers: 4

  Process id: 18287: Average Time: 0.000224 Readers: 6 Writers: 4

  Process id: 18291: Average Time: 0.000272 Readers: 6 Writers: 4

  Process id: 18289: Average Time: 0.000177 Readers: 6 Writers: 4

  Process id: 18293: Average Time: 0.000255 Readers: 6 Writers: 4

  Process id: 18294: Average Time: 0.000232 Readers: 6 Writers: 4

  For entry number: 0
  Total Reads: 20 Total Writes:21

  For entry number: 1
  Total Reads: 40 Total Writes:19

Και τέλος,για την μεταγλώττιση ,εκτέλεση και διαγραφή του εκτελέσιμου:

$ make
$ ./main [όρισμα1] [όρισμα2] [όρισμα3]
$ clean