ΤΑΥΤΟΧΡΟΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ 1

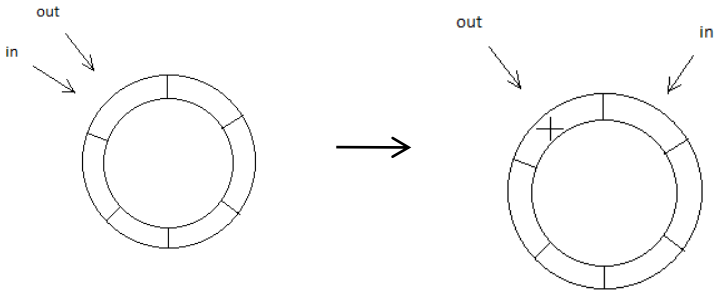
Ομάδα 8

Γιαννούκος Τριαντάφυλλος Ανάργυρος

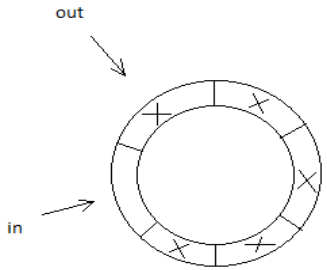
Ματζώρος Χρήστος Κωνσταντίνος

1.1 FIFO Pipe

- **Circular Buffer (in)**
  - Μέσω της pipe_write() γίνεται εγγραφή στη θέση in mod size.
  - Για να γίνει εγγραφή σε μια θέση του πίνακα (ο δείκτης in δείχνει σε αυτή τη θέση), πρέπει πρώτα να γίνει έλεγχος αν ο δείκτης out της pipe_read() δείχνει στην επόμενη θέση από αυτήν που θέλουμε να γράψουμε(in = (in+1) mod size).
  - Εφόσον ισχύει η παραπάνω συνθήκη και γίνει εγγραφή στη θέση in mod size , ο δείκτης in αυξάνεται κατά ένα.
  - Στη συνέχεια φαίνονται οι οριακές περιπτώσεις. Στην πρώτη περίπτωση θα γίνει εγγραφή στην θέση in αφού (in+1)mod size != (out)mod size ενώ στην δεύτερη περίπτωση δεν ισχύει αυτή η συνθήκη, οπότε δεν θα γίνει εγγραφή.



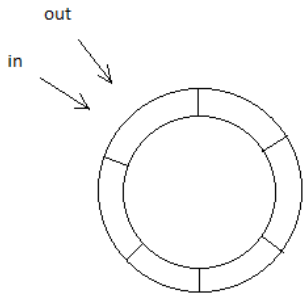
(Σχήμα 1)



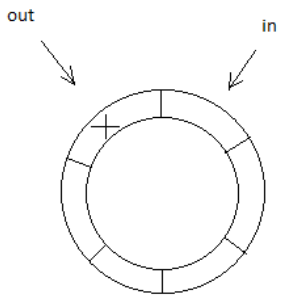
(Σχήμα 2)

1.1 FIFO Pipe

- **Circular Buffer (out)**
  - Μέσω της pipe_read() γίνεται ανάγνωση από τη θέση out mod size.
  - Για να γίνει ανάγνωση το περιεχόμενο από μια θέση του πίνακα, πρέπει πρώτα να γίνει έλεγχος αν ο δείκτης out δείχνει στην ίδια θέση με αυτή που δείχνει ο δείκτης in. Αν ισχύει η παραπάνω συνθήκη δεν θα διαβαστούν δεδομένα από αυτή την θέση και θα έχουμε ενεργή αναμονή μέχρι να γίνει κάποια εγγραφή(δηλαδή ο δείκτης in να προχωρήσει μία θέση).
  - Εφόσον δεν ισχύει η παραπάνω συνθήκη και γίνει ανάγνωση από την θέση out mod size αυξάνω τον δείκτη out κατά 1.
  - Στη συνέχεια φαίνονται οι οριακές περιπτώσεις. Στην πρώτη περίπτωση δεν είναι εφικτή η ανάγνωση από την θέση out αφού (in)mod size == (out)mod size ενώ στην δεύτερη περίπτωση δεν ισχύει αυτή η συνθήκη οπότε επιτρέπεται η ανάγνωση.



(Σχήμα 1)



(Σχήμα 2)

1.1 FIFO Pipe

- Η main() περνάει μέσω της pthread_create στα 2 νήματα που δημιουργεί μια μεταβήτή struct(buf_info). Το struct περιέχει 5 πεδία:
  - Το **μέγεθος** του αγωγού
  - Το **όνομα** και το **file descriptor** για το αρχείο που θα ανοίξει η συνάρτηση που χρησιμοποιεί το 1ο νήμα(write_foo)
  - Το **όνομα** και το **file descriptor** για το αρχείο που θα ανοίξει η συνάρτηση που χρησιμοποιεί το 2ο νήμα(read_foo)

1.1 FIFO Pipe

- **Σημείο συγχρονισμού:**
  - Η main() πρέπει να περιμένει να τελειώσουν τα 2 νήματα που έχει δημιουργήσει πριν τερματίσει
- **Λύση:**
  - Χρήση του flag **threads_finished**, το οποίο:
    - Αρχικοποιείται με 0
    - Όταν τελειώσει κάποια από τις 2 διεργασίες προστίθεται 1 στην τιμή του threads_finished πριν αυτή τερματίσει
    - Η main είναι σε ενεργή αναμονή, μέχρι να τελειώσουν και τα 2 threads, δηλαδή η τιμή της μεταβλητής threads_finished να γίνει 2

1.1 FIFO Pipe

- **Σημείο συγχρονισμού:**
  - Όταν καλείται η pipe_close() και ο αγωγός είναι άδειος πρέπει η pipe_read() να επιστρέψει 0 χωρίς να γίνει ανάγνωση
- **Λύση:**
  - Όταν καλείται η pipe_close() η global μεταβλητή is_pipe_closed γίνεται 1
  - Η pipe_read() όταν διαβάσει όλα τα δεδομένα που είναι γραμμένα στον αγωγό θα μπεί σε ενεργή αναμονή, περιμένοντας:
    - Είτε την είσοδο δεδομένων στον αγωγό για να τα διαβάσει
    - Είτε την αλλαγή της τιμής της μεταβλητής is_pipe_closed σε 1(που σημαίνει ότι έχει ήδη κληθεί η pipe_close())
  - Στην τελευταία περίπτωση η pipe_read() θα επιστρέψει 0 και η read_foo() με τη σειρά της ειδοποιεί τη main και τερματίζει

1.2 Παράλληλος υπολογισμός fractals

- Η main περνάει μέσω της pthread_create στο νήμα που δημιουργεί μια μεταβήτή struct.
- Το struct parameters(worker) περιέχει 4 πεδία:
  - Τη **μεταβλητή struct mandel_Pars** που περιέχει τις παραμέτρους για τον υπολογισμό των fractals
  - Τον **μέγιστο αριθμό των επαναλήψεων**
  - Έναν **pointer**, ο οπόιος διαπερνά τον πίνακα res με τα αποτελέσματα του υπολογισμού των fractals
  - Ένα **flag(status)** για το συγχρονισμό των νημάτων το οποίο παίρνει 4 διαφορετικές τιμές ανάλογα με το στάδιο της διαδικασίας και αρχικοποιείται με 0

1.2 Παράλληλος υπολογισμός fractals

- **Σημείο συγχρονισμού:**
  - Η main() πρέπει αφού αναθέσει μια δουλειά σε έναν worker πρέπει να τον ειδοποιήσει για να αρχίσει τη δουλειά
- **Λύση:**
  - Η main() αφού αναθέσει τη δουλειά αλλάζει τη μεταβλητή **status** σε **1**
  - Ο worker είναι σε ενεργή αναμονή μέχρι η μεταβλητή status να γίνει 1, ώστε να αρχίσει τον υπολογισμό

1.2 Παράλληλος υπολογισμός fractals

- **Σημείο συγχρονισμού:**
  - Η worker πρέπει με τη σειρά του αφού ολοκληρώσει τον υπολογισμό και αποθηκεύσει τα αποτελέσματα να ειδοποιήσει τη main()
- **Λύση:**
  - Ο worker μόλις τελειώσει τη δουλειά αλλάζει τη μεταβλητή **status** σε **2**
  - Η main() είναι σε ενεργή αναμονή μέχρι η μεταβλητή status να γίνει 2, ώστε να σχεδιάσει τα αποτελέσματα του υπολογισμού που πραγματοποίησε ο worker

1.2 Παράλληλος υπολογισμός fractals

- **Σημείο συγχρονισμού:**
  - Η main() οφείλει να περιμένει να τελειώσουν όλοι οι workers για να ξεκινήσει ξανά από την αρχή
- **Λύση:**
  - Μόλις ολοκληρωθεί η εκτύπωση των δεδομένων ενός worker η main() αλλάζει τη μεταβλητή **status** που αντιστοιχεί σε αυτόν σε **3**
  - Όταν τα flags status που αντιστοιχούν σε όλους τους workers πάρουν την τιμή 3, σημαίνει ότι όλα τα αποτελέσματα αυτής της επανάληψης έχουν ολοκληρωθεί και σχεδιαστεί, και άρα η main() είναι έτοιμη να αρχίσει ξανά την επανάληψη

1.3 Παράλληλο Quicksort

- Η main περνάει μέσω της pthread_create στα 2 νήματα που δημιουργεί μια μεταβήτή struct.
- Το struct info περιέχει 4 πεδία:
  - Το **μέγεθος του πίνακα**
  - Τη **θέση του 1ου στοιχείου** προς ταξινόμηση
  - Τη **θέση του τελευταίου στοιχείου** προς ταξινόμηση
  - Ένα **flag(done)** για το συγχρονισμό των νημάτων

1.3 Παράλληλο Quicksort

- **Σημείο συγχρονισμού:**
  - Η main() πρέπει να περιμένει τα τελειώσουν τα 2 νήματα που έχει δημιουργήσει(ένα για το αριστερό κι ένα για το δεξί τμήμα του πίνακα) πριν τερματίσει
- **Λύση:**
  - Τα νήματα πριν τερματίσουν αλλάζουν την τιμή της μεταβλητής τους done σε 1
  - Η main είναι σε ενεργή αναμονή, μέχρι να τελειώσουν και τα 2 threads, δηλαδή οι τιμές και των 2 μεταβλητών done που αντιστοιχούν στα 2 νήματα να γίνουν 1

1.3 Παράλληλο Quicksort

- **Σημείο συγχρονισμού:**
  - Όπως και η main(), έτσι ακριβώς και το κάθε νήμα πρέπει να περιμένει τα τελειώσουν τα 2 επιμέρους νήματα που έχει δημιουργήσει(ένα για το αριστερό κι ένα για το δεξί τμήμα του πίνακα) πριν τερματίσει
- **Λύση**(με τον ίδιο τρόπο που αναφέρθηκε προηγουμένως):
  - Τα νέα νήματα πριν τερματίσουν αλλάζουν την τιμή της μεταβλητής τους **done** σε **1**
  - Το νήμα «πατέρας» είναι σε ενεργή αναμονή, μέχρι να τελειώσουν και τα 2 threads που δημιούργησε, δηλαδή η τιμή και των 2 μεταβλητών done που αναφέρονται σε αυτά να γίνουν 1