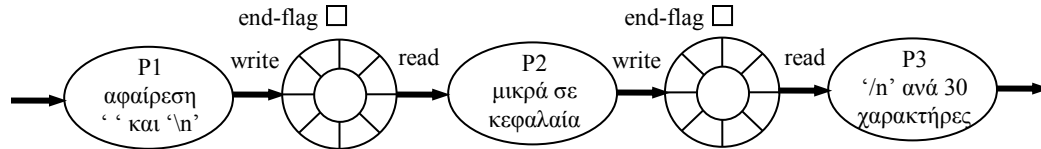


Σειρά Εργασιών 1

1.1 Μορφοποίηση κειμένου ως ροή επεξεργασίας

Αναπτύξτε ένα πρόγραμμα που χρησιμοποιεί 3 νήματα P1, P2 και P3 για να μετασχηματίσει το κείμενο που διαβάζει από την είσοδο του μέσα από μια ροή επεξεργασίας, ως εξής. Το P1 διαβάζει το κείμενο, αφαιρεί τα περιττά κενά και τα 'n' ανάμεσα στις λέξεις, και προωθεί το κείμενο στο P2. Το P2 μετατρέπει τα μικρά γράμματα σε κεφαλαία, και προωθεί το κείμενο στο P3. Τέλος, το P3 εκτυπώνει το κείμενο σε γραμμές των 30 χαρακτήρων.



Η μετάδοση των χαρακτήρων ανάμεσα σε κάθε ζεύγος νημάτων (P1 → P2 και P2 → P3) γίνεται μέσω ενδιάμεσων αποθηκών FIFO που μπορεί να υλοποιηθούν με την τεχνική της «κυκλικής αποθήκης», χωρίς να προκύπτουν προβλήματα ανταγωνισμού. Αν η αποθήκη γεμίσει, το νήμα που γράφει πρέπει να περιμένει μέχρι να δημιουργηθεί χώρος. Αν η αποθήκη είναι άδεια, το νήμα που διαβάζει πρέπει να περιμένει μέχρι να γραφτούν νέοι χαρακτήρες. Επίσης, όταν το νήμα που γράφει γνωρίζει ότι δεν πρόκειται να γράψει άλλους χαρακτήρες, σηματοδοτεί το τέλος της ροής των δεδομένων στο νήμα που διαβάζει για να μην περιμένει ατέρμονα. Το πρόγραμμα πρέπει να τερματίζει αφού ολοκληρωθεί ο μετασχηματισμός και η εκτύπωση ολόκληρου του κειμένου.

Μετρήστε και σχολιάστε την ταχύτητα επεξεργασίας (για σχετικά μεγάλα κείμενα) ως συνάρτηση του μεγέθους των ενδιάμεσων αποθηκών, π.χ., με χωρητικότητα 10, 100 και 1000 χαρακτήρων.

1.2 Παράλληλος υπολογισμός fractals

Στην σελίδα του μαθήματος δίνεται έτοιμο ένα πρόγραμμα που υπολογίζει και σχεδιάζει το Mandelbrot set. Το πρόγραμμα υποδιαιρεί την περιοχή σε N τμήματα, και υπολογίζει κάθε τμήμα ξεχωριστά, σε βρόγχο.

Αλλάξτε τον κώδικα έτσι ώστε κάθε τμήμα να υπολογίζεται από ένα ξεχωριστό νήμα «εργάτη». Όταν ένας εργάτης ολοκληρώνει τον υπολογισμό του, το κυρίως νήμα σχεδιάζει το αποτέλεσμα. Ο υπολογισμός της επόμενης περιοχής αρχίζει αφού ολοκληρωθεί αυτός της προηγούμενης. Το πρόγραμμα πρέπει να χρησιμοποιεί τους ίδιους εργάτες (όχι να δημιουργεί νέους εργάτες για κάθε υπολογισμό). Η λογική του κυρίως νήματος και ενός νήματος εργάτη δίνεται ενδεικτικά παρακάτω:

<pre> main: create N worker threads while (new problem input received) { create N jobs and assign to workers notify workers to start computing while (not all workers done) { wait for a worker to finish job read & draw the result } } </pre>	<pre> worker: while (1) { wait for main to assign job read job parameters perform the computation store result notify main that job is finished } </pre>
--	---

Μετρήστε και σχολιάστε την επιτάχυνση που επιτυγχάνεται ως συνάρτηση του N, π.χ. 1, 10, 100, καθώς και της «ακρίβειας» με την οποία γίνεται ο υπολογισμός.

Σημείωση για τις παραπάνω εργασίες: Η υλοποίηση πρέπει να γίνει σε C με pthreads. Ο συγχρονισμός μεταξύ των νημάτων πρέπει να υλοποιηθεί με απλές-κοινές μεταβλητές και ενεργή αναμονή, χωρίς να χρησιμοποιηθεί κάποιος από τους μηχανισμούς συγχρονισμού των pthreads.