Σειρά Εργασιών 3

3.1 Αναγνώριση πρώτων αριθμών (με ελεγκτή)
Τροποποιήστε τη λύση που αναπτύξατε στην εργασία 2.2 έτσι ώστε ο επιθυμητός συγχρονισμός να επιτυγχάνεται στο πνεύμα ενός ελεγκτή (monitor).

3.2 Στενή γέφυρα (με ελεγκτή)
Τροποποιήστε τη λύση που αναπτύξατε στην εργασία 2.3 έτσι ώστε ο επιθυμητός συγχρονισμός να επιτυγχάνεται στο πνεύμα ενός ελεγκτή (monitor). Δοκιμάστε την υλοποίηση σας μέσω ενός κατάλληλου προγράμματος προσομοίωσης (σαν και αυτό που ήδη φτιάξατε για την εργασία 2.3).

3.3 Τρενάκι (με ελεγκτή)
Τροποποιήστε τη λύση που αναπτύξατε στην εργασία 2.4 έτσι ώστε ο επιθυμητός συγχρονισμός να επιτυγχάνεται στο πνεύμα ενός ελεγκτή (monitor). Δοκιμάστε την υλοποίηση σας μέσω ενός κατάλληλου προγράμματος προσομοίωσης (σαν και αυτό που ήδη φτιάξατε για την εργασία 2.4).

3.4 Συγχρονισμός στο πνεύμα των conditional critical regions
Θέλουμε να υποστηρίξουμε τον συγχρονισμό νημάτων στο πνεύμα των conditional critical regions (CCRs) μέσω των εξής ενδεικτικών εντολών:

CCR_DECLARE(label)	Δήλωση πλαισίου συγχρονισμού με (μοναδικό) όνομα label. Υποθέστε ότι αυτή η εντολή εμφανίζεται σε καθολικό επίπεδο προγράμματος.
CRR_INIT(label)	Αρχικοποίηση του πλαισίου συγχρονισμού με όνομα label. Υποθέστε ότι αυτή η εντολή εμφανίζεται σε κατάλληλο σημείο του προγράμματος, προτού δημιουργηθούν νήματα που θα χρησιμοποιούν το πλαίσιο συγχρονισμού.
CCR_DO(label, lexpr, body)	Εκτέλεση του κώδικα body υπό την λογική συνθήκη lexpr στο πλαίσιο συγχρονισμού με όνομα label. Αν η συνθήκη δεν είναι αληθής, το νήμα που καλεί στην εντολή μπλοκάρει. Υποθέστε ότι αυτή η εντολή εμφανίζεται μέσα στον κώδικα που εκτελούν τα ανταγωνιστικά νήματα.

Ως πρώτο βήμα, σχεδιάστε μια λύση στο χαρτί: σκεφτείτε τι δηλώσεις μεταβλητών και τι κώδικας χρειάζεται για να υλοποιηθεί κάθε μια από τις παραπάνω εντολές. Βεβαιωθείτε για την ορθότητα της λύσης σας προτού προχωρήσετε στο επόμενο βήμα.

Ως δεύτερο βήμα, υλοποιήσετε τη λύση σας μέσω κατάλληλων μακροντετολών για τον C preprocessor (ή για έναν δικό σας προεπεξεργαστή) που θα κάνει τις αντίστοιχες αντικαταστάσεις στον πηγαίο κώδικα προγραμμάτων που χρησιμοποιούν τις παραπάνω εντολές. Ορίστε μια κατάλληλη σύμβαση για την ονομασία των μεταβλητών που εισάγονται (μέσω των μακροεντολών) για κάθε πλαίσιο συγχρονισμού, έτσι ώστε να μην υπάρχει σύγκρουση ονομάτων με τις υπόλοιπες «κανονικές» μεταβλητές του προγράμματος. Η υλοποίησή σας πρέπει να είναι σχεδιασμένη έτσι ώστε το ένα πρόγραμμα να μπορεί να χρησιμοποιεί πολλά διαφορετικά/ανεξάρτητα πλαίσια συγχρονισμού.

Ως τρίτο βήμα, σκεφτείτε και υλοποιήστε νέες λύσεις για τις εργασίες 3.1 και 3.2 με βάση τις παραπάνω μακροεντολές, και δοκιμάστε τη λειτουργικότητα τους. Και αυτός ο κώδικας αποτελεί παραδοτέο.

Η υλοποίηση των εργασιών 3.1, 3.2 και 3.3 μπορεί να γίνει χρησιμοποιώντας συνδυαστικά mutexes και conditions της βιβλιοθήκης pthread. Εναλλακτικά, μπορείτε να γράψετε την υλοποίηση σε Java χρησιμοποιώντας την λειτουργικότητα synchronized και wait/notify που προσφέρει η γλώσσα.

Η υλοποίηση της εργασίας 3.4 πρέπει να γίνει σε C χρησιμοποιώντας συνδυαστικά mutexes και conditions της βιβλιοθήκης pthread.

Παράδοση: Σάββατο 19 Δεκεμβρίου 2020, 23:59