Άσκηση 2: Συγχρονισμός εκτέλεσης νημάτων

Μαρία-Νίκη Ζωγράφου
ΑΜ:1096060

Περιγραφή:
Να γραφεί πρόγραμμα σε γλώσσα C, που θα συγχρονίζει την εκτέλεση των παρακάνω νημάτων με χρήση σημαφόρων, ώστε το νήμα 1 να εμφανίζει τη σωστή τιμή του x.

δηλ. έχουμε 3 νήματα t1, t2, t3 με κάποιες κοινές μεταβλητές (a2, b1, c1, c2, x, y, z) που κάνουν τα εξής:

t1: a1 = 10
    a2 = 11
    y = a1 + c1
    print(x)

t2: b1 = 20
    b2 = 21
    w = b2 + c2
    x = z - y + w

t3: c1 = 30
    c2 = 31
    z = a2 + b1

προτείνετε μια λύση με το μικρότερο αριθμό σημαφόρων.

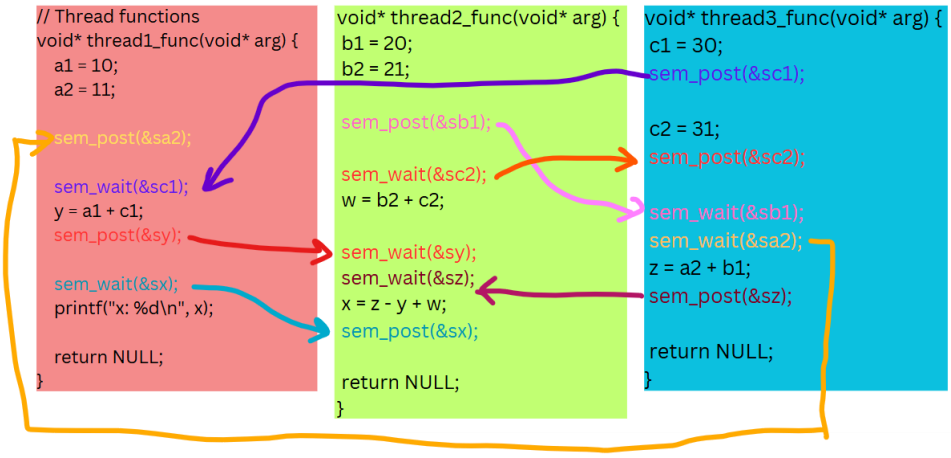
Τρόπος Σκέψης:

Πρέπει να φροντίσουμε πώς πριν χρησιμοποιηθεί κάποια από τις 10 κοινές μεταβλητές (int a1, a2, b1, b2, c1, c2, y, w, x, z;) έχει γίνει ο υπολογισμός της. Αυτό μπορεί να επιτευχθεί με σεμαφόρους. Σε περίπτωση που κάποιο resource δεν έχει αρχικοποιηθεί θα μπουν σε ουρά αναμονής.

Παρατηρούμε επίσης ότι οι ακέραιοι a1,b2 χρησιμοποιούνται μόνο από το ίδιο νήμα που τους αρχικοποίησε, επομένως δεν χρειάζεται να έχουν σεμαφόρο.

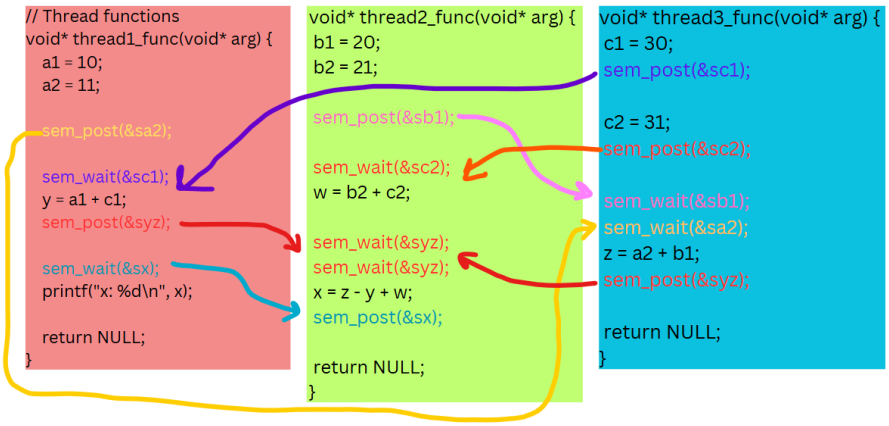
1ο βήμα:

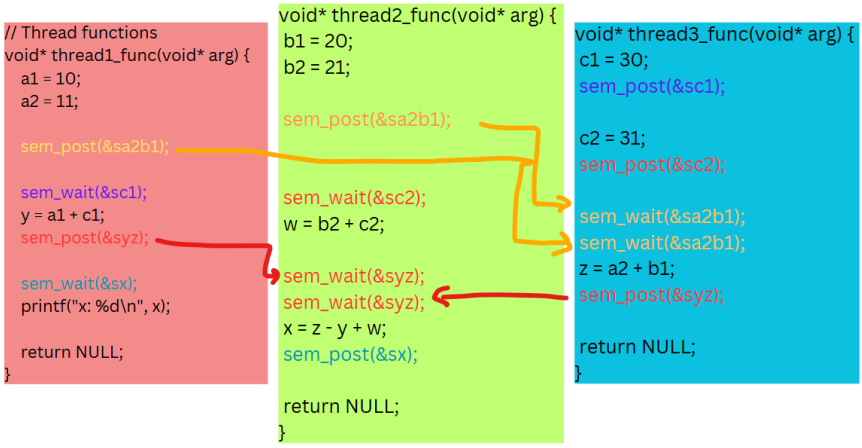
Για κάθε resource-μεταβλητή χρησιμοποιούμε έναν σεμαφόρο- σύνολο 7.



2ο βήμα:

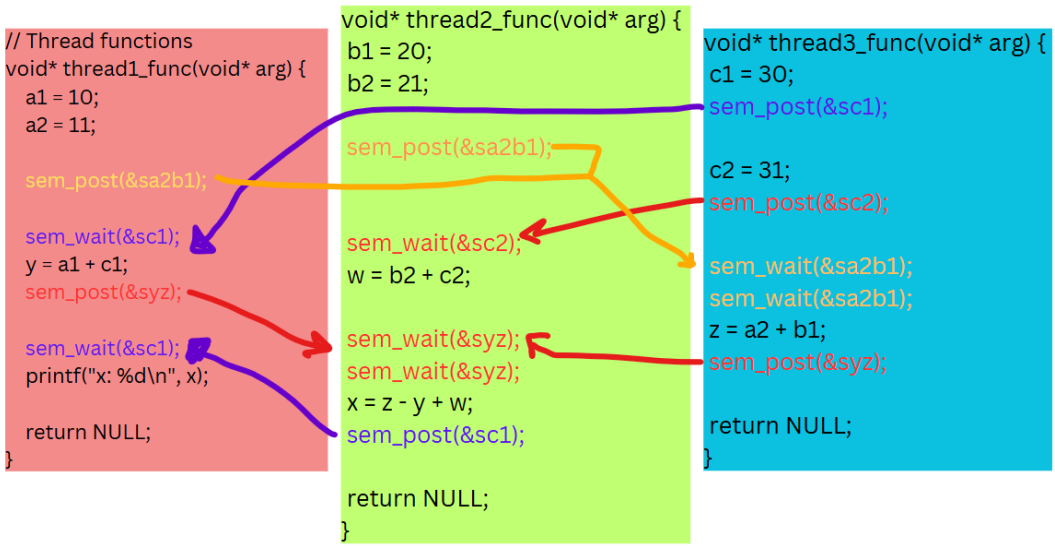
Οι μεταβλητές y και z χρησιμοποιούνται στην ίδια εντολή. Αντίστοιχα και οι μεταβλητές a2,b1. Μπορούμε λοιπόν να χρησιμοποιήσουμε έναν κοινό σεμαφόρο για αυτά τα δύο resources και να καλέσουμε την wait δυο φορές, ώστε να είμαστε σίγουροι ότι έχουν γίνει οι δύο απαραίτητες πράξεις προηγουμένως. Επιτυγχάνεται μείωση σεμαφόρων σε 5.





3ο βήμα:

Επαναχρησιμοποιήση σεμαφόρων: Παρατηρούμε ότι ο sc1 θα μπορούσε να χρησιμοποιηθεί και για τον x αφού τα δυο wait καλούνται στο ίδιο thread διαδοχικά. Πρώτα καλείται για να χρησιμοποιηθεί το c1. Υπάρχει περίπτωση να έχει υπολογιστεί το x και όχι το c1 και να μην μπει σε ουρά το thread1; Προκειμένου να γίνει post το x χρειάζεται σίγουρα να έχει κληθεί πρώτα το thread3 και να έχει τρέξει όλο, οπότε αν



έχει υπολογιστεί το x έχει υπολογιστεί πρώτα και το c1. Άρα μπορεί να γίνει χρήση μόνο ενός σεμαφόρου για τα x και c1.

4ο βήμα:

Μπορούμε να χρησιμοποιήσουμε τον ίδιο σεμαφόρο αντί για sc2 και syz. Αν όμως αφήσουμε τον κώδικα έτσι όπως είναι υπάρχει η περίπτωση να υπολογιστεί y και να γίνει sem_post(yz) και ύστε να προσπαθήσει να υπολογίσει το w=b2+c2 χωρίς το c2 να έχει υπολογιστεί! Π.χ. Κλήση thread3 και υπολογισμός μόνο του c1, ύστερα κλήση του thread 1 και υπολογισμός του y και τέλος κλήση thread2 και προσπάθεια υπολογισμού του w. Οπότε αναδιατάσουμε την σειρά των εντολών, ώστε να είμαστε σίγουροι ότι τα y,z,c2 έχουν υπολογιστεί όλα με οποιαδήποτε σειρά και ύστερα να υπολογιστούν τα w και x. ΄Ετσι μένουν 3 σεμαφόροι στο τελικό μας πρόγραμμα.