

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΦΡΟΝΤΙΣΤΗΡΙΟ – ΑΣΚΗΣΗ 2

ΜΠΑΣΑΓΙΑΝΝΗ ΓΕΩΡΓΙΑ

1084016

Για την εκπόνηση της άσκησης τα αρχεία με τους κώδικες στη c γράφτηκαν με χρήση VSCode και για την μεταγλώττιση και εκτέλεση τους χρησιμοποιήθηκε το WSL (μέσω terminal).

Συγχρονισμός εκτέλεσης νημάτων - Λύση

Δημιουργία 3 thread με κοινές μεταβλητές μεταξύ τους συγχρονισμένα, έτσι ώστε με τη βοήθεια σημαφόρων να τυπώνει το νήμα 1 την σωστή τιμή του x.

Thread 1: $a1=10$, $a2=11$, $y=a1+c1$, $\text{print}(x)$

Thread 2: $b1=20$, $b2=21$, $w=b2+c2$, $x=z-y+w$

Thread 3: $c1=30$, $c2=31$, $z=a2+b1$

Για τον υπολογισμό της σωστής τιμής του x, θα πρέπει πρώτα να ανατεθούν τιμές στις μεταβλητές $a1, a2, b1, b2, c1, c2$, ώστε να υπολογιστούν σωστά πρώτα τα z, y, w και τέλος το x.

Η λύση που προτείνω είναι με 3 σημαφόρους.

```
C f2.c X
C: > Users > georg > OneDrive - University of Patras > Desktop > mathimata > 7o > OS > frontistiriakes > C f2.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <pthread.h>
4  #include <semaphore.h>
5
6  int a1,a2,b1,b2,c1,c2,y,z,x,w;
7  sem_t sem1, sem2, sem3;
8
9  void* f1()
10 {
11     sem_wait(&sem2);
12     a1=10;
13     a2=11;
14     y=a1+c1;
15     // printf("y=%d\n", y);
16     sem_post(&sem3);
17
18     sem_wait(&sem2);
19     printf("x=%d\n", x);
20     sem_post(&sem2);
21 }
22
23 void* f2()
24 {
25     sem_wait(&sem3);
26     b1=20;
27     b2=21;
28     w=b2+c2;
29     sem_post(&sem1);
30
31     sem_wait(&sem3);
32     x=z-y+w;
33     // printf("w=%d  x=%d\n", w,x);
34     sem_post(&sem2);
35 }
```

```

36
37 void* f3()
38 {
39     sem_wait(&sem1);
40     c1=30;
41     c2=31;
42     sem_post(&sem2);
43
44     sem_wait(&sem1);
45     z=a2+b1;
46     // printf("z=%d\n", z);
47     sem_post(&sem3);
48 }
49
50 int main()
51 {
52     //initialize an UNNAMED semaphore to 1
53     sem_init(&sem1, 0, 1);
54     sem_init(&sem2, 0, 0);
55     sem_init(&sem3, 0, 0);
56     pthread_t thread1, thread2, thread3;
57
58     //create independent threads to execute f1, f2
59     pthread_create(&thread1, NULL, f1, NULL);
60     pthread_create(&thread2, NULL, f2, NULL);
61     pthread_create(&thread3, NULL, f3, NULL);
62
63     //wait until threads are completed before main thread continues
64     pthread_join(thread1, NULL);
65     pthread_join(thread2, NULL);
66     pthread_join(thread3, NULL);
67
68     //destroy the UNNAMED semaphore
69     sem_destroy(&sem1);
70     sem_destroy(&sem2);
71     sem_destroy(&sem3);
72     return 0;
73 }

```

Όταν δημιουργηθούν τα νήματα το καθένα θα εκτελέσει τη συνάρτησή του. Η f1 και η f2 θα μπουν σε ουρά αναμονής για τις σημαφόρους sem2 και sem3 αντίστοιχα {sem_wait(&sem2), sem_wait(&sem3)}, καθώς στην main έχουν αρχικοποιηθεί στην τιμή 0. Ενώ η f3 θα μπει στο critical path επειδή η sem1 έχει αρχικοποιηθεί στην τιμή 1 {sem_wait(&sem1), με την τιμή της sem1 να γίνεται τώρα 0}.

Επιλέχθηκε να ξεκινήσει το thread3 επειδή τις μεταβλητές c1 και c2, τις χρειάζονται τα thread1 και thread2 αντίστοιχα για να υπολογίσουν τα γ, w μαζί με μία «δική τους» μεταβλητή.

Έπειτα αφού αρχικοποιηθούν οι c1, c2, αυξάνεται η τιμή της sem2 {sem_post(&sem2)} και πλέον το thread1 μπορεί να βγει από την ουρά αναμονής, να διαβάσει τις a1, a2 και να υπολογίσει τη σωστή τιμή του γ=40(10+30). Αυξάνεται η τιμή της sem3 {sem_post(&sem3)} και τώρα μπορεί το thread2 να βγει από την ουρά αναμονής, να διαβάσει τις b1, b2 και να υπολογίσει τη σωστή τιμή του w=52(21+31).

Την ίδια στιγμή τα thread3 και thread1 είναι σε ουρά αναμονής για τις σημαφόρους sem1, sem2 {sem_wait(&sem1), sem_wait(&sem2)} αντίστοιχα. Το thread2 μετά τον υπολογισμό του w αυξάνει την τιμή του sem1 {sem_post(&sem1)} κι έτσι βγαίνει το thread3 από την ουρά αναμονής και υπολογίζει την σωστή τιμή του z=31(11+20). Μετά, αυξάνει την τιμή του sem3 {sem_post(&sem3)} και τελειώνει το Thread3.

Την ίδια στιγμή το thread2 ήταν σε ουρά αναμονής {sem_wait(&sem3)} και βγήκε τώρα με το τέλος του thread3 για να υπολογιστεί η σωστή τελική τιμή του x=43(31-40+52). Τέλος, αυξάνει την τιμή του sem2 {sem_post(&sem2)} και βγάζει το thread1 από την ουρά αναμονής για να τυπώσει την σωστή τιμή του x, όπως ακριβώς ζητήθηκε.

(Μάλιστα οι τιμές των y, w, z έχουν μπει σε σχόλιο να εκτυπώνονται για πιο συγκεκριμένο έλεγχο αν χρειαστεί σε περίπτωση που εκτελέσετε τον κώδικα και είναι πάντα οι επιθυμητές τιμές).