

Παπαδόπουλος Παντελής

A.M 1041854

Αναφορά άσκησης 3

Ξεκινώντας την υλοποίηση της άσκησης 3 παρατήρησα πως η συνάρτηση display τυπώνει την έκφραση που της δίνεται γράμμα-γράμμα. Επομένως ήταν απαραίτητος ο συγχρονισμός ή/και ο αμοιβαίος αποκλεισμός μεταξύ των processes και των threads που χρησιμοποιούν την παραπάνω συνάρτηση.

Syn process 1

Εδώ έγινε η χρήση του αμοιβαίου αποκλεισμού μέσω των σημαφόρων. Σαν critical region θεώρησα τη συνάρτηση display. Τα δύο process (parent, child) εκτελούνται σειριακά. Με τη χρήση του

down()

display

up()

καθώς και μιας σημαφόρου όταν εκτελείται το ένα process «λέμε» στο άλλο που προσπαθεί να εκτελεστεί να «περιμένει». Έτσι επιτεύχθηκε το σωστό output χωρίς να μπερδεύονται οι δυο εκφράσεις «Hello world» και «Kalimera kosme».

Syn process 2

Στο συγκεκριμένο κομμάτι ήταν απαραίτητη η χρήση του συγχρονισμού των διεργασιών. Επομένως το process 1 «ab» πρέπει να ξεκινήσει πρώτο ενώ το process 2 «cd\n» οφείλει να περιμένει το πρώτο ώστε να επιτευχθεί το σωστό output. Άρα πριν την εκτέλεση του κάθε process πρέπει να υπάρχει ένα down() για να «περιμένει» σήμα ότι είναι η σειρά του να εκτελεστεί ενώ μετά την εκτέλεση του πρέπει να δώσει το απαραίτητο σήμα up().

Syn thread 1

Η χρήση των threads ήταν διαφορετική και ταυτόχρονα παρόμοια με τη χρήση των processes και των σημαφόρων. Η τεχνική που ακολούθησα ήταν η ίδια. Σαν critical region θώρησα τη συνάρτηση display και χρησιμοποίησα τη λειτουργία lock και unlock του mutex πριν και μετά, αντίστοιχα, από αυτή για να επιτύχω το επιθυμητό output. Το πρόγραμμα ξεκινάει δημιουργώντας 2 threads τα οποία διαχειρίζονται την ίδια συνάρτηση func(η οποία περιέχει το display) αλλά το καθένα περνάει μια διαφορετική μεταβλητή μέσα σε αυτή. Έπειτα εκτελείται το display παρουσιάζοντας το σωστό output ενώ το πρόγραμμα με την εντολή exit τερματίζει τα threads περιμένοντας τα όμως να τελειώσουν τη λειτουργία τους(εντολή pthread_join).

Syn_thread 2

Εδώ δημιουργούνται πάλι 2 threads αλλά χρησιμοποιούνται 2 συναρτήσεις func1 και func2 για την επίτευξη του συγχρονισμού. Η χρήση του mutex ήταν απαραίτητη διότι τα 2 αυτά threads δεν πρέπει να τρέχουν παράλληλα. Βέβαια όπως και στο syn_process_2 οι λειτουργίες signal και wait μέσω των conditional variables ήταν καθοριστικές για την επίτευξη του σωστού output. Αφού εκτελεστεί το πρώτο thread πρέπει να στείλει σήμα στο δεύτερο να ξεκινήσει και αυτό με τη σειρά του να ξαναστείλει σήμα στο πρώτο κ.ο.κ.

-Σε κάθε κώδικα γίνεται πάντα η σωστή διαγραφή των σημαφόρων και ο σωστός τερματισμός των threads, του mutex και του cond-

Προβλήματα

Το κυριότερο πρόβλημα που συνάντησα ήταν η πλήρης κατανόηση των σημαφόρων και των threads. Αφου έψαξα και είδα αρκετά παραδείγματα που περιείχαν συγχρονισμό ή/και αμοιβαίο αποκλεισμό μεταξύ processes και threads η υλοποίηση της άσκησης ήταν θέμα trial and error όσον αφορά την σωστή διάταξη των up και down για τις σημαφόρους, την δημιουργία και τον τερματισμό των threads καθώς και τη χρήση των lock, unlock, wait και signal.