

DOCUMENTATION 3ης ΑΣΚΗΣΗΣ

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΦΟΙΤΗΤΗΣ: ΣΟΥΚΑΡΑΣ ΣΩΤΗΡΗΣ

ΑΜ: 236205

EMAIL: soukaras@ceid.upatras.gr

- **syn_process_1.c:** Στην αρχή δηλώνουμε τις σηματοφόρους που θα χρησιμοποιήσουμε. Με την εντολή `semget` δημιουργούμε τις σηματοφόρους η καθεμία με διαφορετικό `id`. Μέσα στην διεργασία-παιδί κάνουμε UP μπαίνει στην επανάληψη κάνει DOWN γράφει το μήνυμα και μετά UP και περνάει στην διεργασία-πατέρα όπου κάνει DOWN γράφει το δεύτερο μήνυμα κάνει UP και περνάει στη διεργασία-παιδί. Στο τέλος μετά την εναλλάξ εγγραφή των μηνυμάτων αποδεσμεύεται η μνήμη που καταλαμβάνουν οι σηματοφόροι.

*χρησιμοποιούμε την `semctl` και σαν 3ο όρισμα της δίνουμε `IPC_RMID` και έτσι διαγράφουμε το σύνολο των σηματοφόρων αποδεσμεύοντας μνήμη.

- **syn_process_2.c:** Ίδια ακριβώς με την `syn_process_1` με τη μόνη διαφορά ότι επειδή το πρώτο μήνυμα δεν έχει `\n`, το δεύτερο μήνυμα θα γραφτεί στην ίδια σειρά και μετά θα αλλάξει γραμμή.
- **syn_thread_1.c:** Δημιουργούμε 2 threads τα οποία θα εκτυπώσουν στην οθόνη τα δύο μηνύματα χωρίς να μπλεχτούν τα Output του καθενός. Κάθε μήνυμα βρίσκεται σε 2 συναρτήσεις που η κάθε συνάρτηση έχει διαφορετικό `id` από την άλλη. Ανάμεσα στα `lock` και `unlock` που είναι η κρίσιμη περιοχή του κώδικα αρχικά το πρόγραμμα καλεί το πρώτο μήνυμα 10 φορές και μόλις τελειώσει καλεί το δεύτερο 10 φορές και το εμφανίζει. Μετά την επιτυχή εμφάνιση των μηνυμάτων τα `mutexes` καταστρέφονται και αποδεσμεύουν μνήμη.

*χρησιμοποιούμε `mutexes` για το συγχρονισμό των thread. Σε κάθε συνάρτηση πριν εμφανιστεί το μήνυμα γίνεται Lock το mutex και μόλις γραφτεί στο output γίνεται unlock.

- **syn_thread_2.c:** Ίδια σκέψη με την `syn_thread_1` απλά σε αυτή την περίπτωση τα μηνύματα εμφανίζονται ανακατεμένα. Τα μηνύματα είναι αντίστοιχα σε δυο συναρτήσεις. Κάθε φορά που καλείται μια συνάρτηση εμφανίζεται το μήνυμα κάνει `wait` και πάει στη δεύτερη συνάρτηση, εμφανίζει το μήνυμα της και κάνει `signal` για να τρέξει τη συνάρτηση για το πρώτο μήνυμα. Η όλη διαδικασία θα τελειώσει μόλις ο buffer γίνει 19 . κάθε φορά που γράφεται ένα μήνυμα ο buffer αυξάνεται κατά 1. Επειδή πρέπει ν εμφανιστεί 10 φορές

η λέξη `abcd` αυτό σημαίνει ότι θα εμφανιστεί 10 φορές η λέξη `ab` και 10 η `cd` εναλλάξ όμως για αυτό το `buffer` πρέπει ν γίνει 19 για να σταματήσει αφού ξεκινά από τη θέση 0.

*επειδή πρέπει να εμφανίζεται `abcd` σε κάθε γραμμή, γίνεται χρήση των μεταβλητών συνθήκης σε συνεργασία με τα `mutexes`. Κάθε φορά που καλείται η πρώτη συνάρτηση που περιέχει το `ab` μόλις αυτό γραφτεί στην οθόνη καλεί την `pthread_cond_wait` ώστε να περιμένει να δώσει `signal` στη δεύτερη συνάρτηση που περιέχει το `cd`. Μόλις γράψει και το `cd` και αλλάξει γραμμή ειδοποιεί την πρώτη συνάρτηση να γράψει το μήνυμα και συνεχίζεται αυτή η διαδικασία μέχρι να τυπωθεί 10 φορές η λέξη `abcd`.