**Άσκηση 1**

Δίνεται ένα σύστημα με 3 παραγωγούς και ένα καταναλωτή.

Κάθε διεργασία παραγωγός έχει ένα κρίσιμο τμήμα που τοποθετεί ένα στοιχείο σε μία μνήμη. Ο καταναλωτής έχει ένα κρίσιμο τμήμα που καταναλώνει τα *τρία* στοιχεία που τοποθετούν οι παραγωγοί. Ο καταναλωτής μπορεί να μπει σε κρίσιμο τμήμα μόνο όταν εκτελεστούν και οι τρεις παραγωγοί. Οι παραγωγοί (εκτός της πρώτης εκτέλεσης) μπαίνουν σε κρίσιμο τμήμα με τη σειρά Παραγωγός 1, Παραγωγός 2, Παραγωγός 3 αφότου καταναλωθούν τρία στοιχεία από τον καταναλωτή.

Producer 1 Producer 2 Producer 3 Consumer

: : : :

: : : :

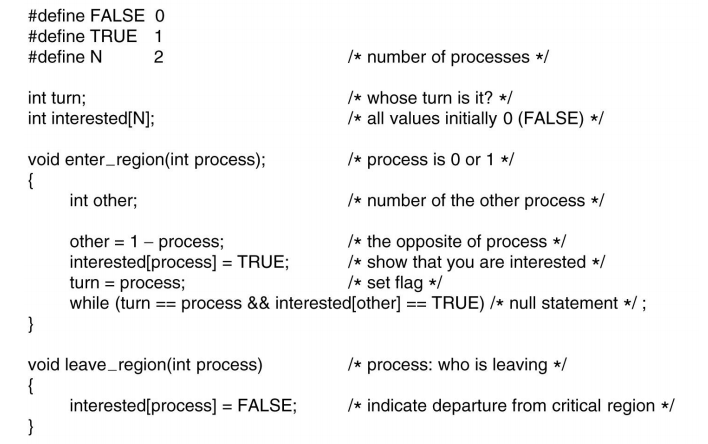
AddItem() AddItem() Add(item) ConsumeItem()

: : : :

Να ορίσετε σηματοφορείς για το παραπάνω σύστημα.

**Άσκηση 2**

Έστω ότι δύο διεργασίες Δ1 και Δ0 τρέχουν με σταθερά κβάντα 100 χρονικές μονάδες και χρησιμοποιούν τη λύση του Peterson, όπως φαίνεται παρακάτω. Έστω ότι ξεκινάει αρχικά η Δ0 και οι 100 μονάδες αρκούν μέχρι να τεθεί interested[process]=true (πριν αλλάξει η turn). Για τη Δ1, οι 100 μονάδες αρκούν για να μπει στο βρόχο while της εισαγωγής στο κρίσιμο τμήμα. Σε ποια χρονική στιγμή θα μπουν σε ΚΤ οι δύο διεργασίες (αν θα μπουν); Ξεκινήστε από το χρόνο t=0.

****

**Άσκηση 3**

Να γράψετε μία επέκταση του προβλήματος παραγωγού καταναλωτή, στο οποίο υπάρχουν 2 καταναλωτές και ισχύουν οι εξής κανόνες:

1) Ο παραγωγός πρέπει να γράφει στη μνήμη αφότου γίνει ανάγνωση από τους καταναλωτές, με εξαίρεση την πρώτη εγγραφή

2) Κάθε στοιχείο που παράγεται πρέπει να καταναλώνεται και από τους δύο καταναλωτές με τη σειρά Καταναλωτής 1, Καταναλωτής 2

3) Οι καταναλωτές δεν μπορούν να διαβάζουν ταυτόχρονα

Process Producer Process Consumer 1 Process Consumer 2

: : :

: : :

repeat repeat repeat

add\_item consume\_item consume\_item

: : :

: : :

until forever until forever until forever

**Άσκηση 4**

Με δεδομένη την (γενικά εσφαλμένη) παραδοχή, ότι καθεμία από τις διεργασίες θα λαμβάνει τα κατάλληλα κβάντα, έτσι ώστε να εκτελεστεί ως το τέλος της (δηλαδή ΔΕΝ θα κοπεί στη μέση), θεωρήστε ένα σύστημα, το οποίο εκτελεί 5 διεργασίες. Να ορίσετε τους σηματοφορείς και τον τρόπο χρησιμοποίησης των λειτουργιών up/down (χωρίς να αλλάξετε με άλλον τρόπο τον κώδικα), έτσι ώστε να είναι δυνατή η επαναληπτική εκτύπωση της συμβολοσειράς

ΑDBBBBCCCEE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Διεργασία 1 | Διεργασία 2 | Διεργασία 3 | Διεργασία 4 | Διεργασία 5 |
| .... | .... | .... | .... | .... |
| Επανάλαβε | Επανάλαβε | Επανάλαβε | Επανάλαβε | Επανάλαβε |
| .... | .... | .... | .... | .... |
| Τύπωσε Α | Τύπωσε Β | Τύπωσε Γ | Τύπωσε Δ | Τύπωσε Ε |
| ..... | ..... | ..... | ..... | ..... |
| Έως (για πάντα) | Έως (για πάντα) | Έως (για πάντα) | Έως (για πάντα) | Έως (για πάντα) |

**Άσκηση 5**

Δίνονται 2 διεργασίες εκτύπωσης, οι οποίες εκτελούνται ανά 200 ms και χρησιμοποιούν TSL. Ο κώδικας του τμήματος εκτύπωσης (κρίσιμο τμήμα) είναι ο εξής:

Διάβασε τη μεταβλητή In

Γράψε στο Spool Directory

Θέσε In=In+1

Έστω ότι τα κβάντα και των 2 διεργασιών φτάνουν για να διαβάσουν τη μεταβλητή In, η οποία πριν την εκκίνηση της εκτέλεσης των διεργασιών έχει τιμή 5. Να δώσετε τις τιμές των Flag, Reg και In, όταν μία από τις διεργασίες καταφέρει να μπει σε κρίσιμο τμήμα.

**Άσκηση 6**

Να γράψετε μία επέκταση του προβλήματος παραγωγού- καταναλωτή, στο οποίο υπάρχουν 2 παραγωγοί και 3 καταναλωτές και η σειρά εκτέλεσης είναι: Π1, Π2 και τυχαία σειρά κατανάλωσης. Μόλις καταναλώσουν όλοι οι καταναλωτές, αρχίζει ξανά ο Π1.

**Άσκηση 7**

Να εξετάσετε πιθανά σενάρια διακοπής των κβάντων όταν χρησιμοποιείται η λύση Peterson και TSL. Να ελέγξετε τις περιπτώσεις ενεργού αναμονής και αμοιβαίου αποκλεισμού.