

3^η Σειρά Ασκήσεων

Παράδοση: Τρίτη 22/10/2013

Άσκηση 1

Δείξτε ότι ο αλγόριθμος 2-επεξεργαστών με την χρήση της μεταβλητής Priority ικανοποιεί τον Αμοιβαίο Αποκλεισμό και αποφεύγει το Αδιέξοδο και την Παρατεταμένη Στέρηση.

Άσκηση 2

Στην επόμενη σελίδα παρουσιάζεται ο αλγόριθμος αμοιβαίου αποκλεισμού του Peterson που εγγυάται αποφυγή παρατεταμένης στέρησης. Για τους Αλγόριθμους 2 και 3 που επίσης φαίνονται στην επόμενη σελίδα, δείξτε αν ισχύουν ή όχι τα ακόλουθα:

1. Εγγυάται Αμοιβαίο Αποκλεισμό
2. Αποφεύγει το Αδιέξοδο
3. Αποφεύγει την Παρατεταμένη Στέρηση

Άσκηση 3

Δείξτε γιατί ο αλγόριθμος του Lamport διασφαλίζει τον Αμοιβαίο Αποκλεισμό μόνο αν τα κανάλια επικοινωνίας ικανοποιούν την στρατηγική FIFO. Συγκεκριμένα θεωρήστε την έκδοση του αλγορίθμου που δεν ικανοποιεί την συνθήκη αυτή και κατασκευάστε μια εκτέλεση του αλγορίθμου αυτού που παραβιάζει τον Αμοιβαίο Αποκλεισμό.

Άσκηση 4

Ένας αλγόριθμος λύνει το πρόβλημα του 3-Αμοιβαίου Αποκλεισμού αν σε κάθε στιγμή το πολύ τρεις επεξεργαστές βρίσκονται στο κρίσιμο τμήμα. Γράψετε ένα αλγόριθμο που να λύνει το πρόβλημα του 3-Αμοιβαίου Αποκλεισμού χρησιμοποιώντας read-modify-write κοινόχρηστες μεταβλητές. Δείξτε κάτω ότι ο αλγόριθμος σας αποφεύγει το Αδιέξοδο και την Παρατεταμένη Στέρηση. Κάτω από ποιες προϋποθέσεις συμβαίνει αυτό;

Algorithm 1 Peterson's algorithm

Initially flag[0] and flag[1] are false and turn is 0

code for p_0	code for p_1
⟨Entry⟩:	⟨Entry⟩:
1: flag[0]:=true	1: flag[1]:=true
2: turn:=1	2: turn:=0
3: while flag[1] and turn=1 do wait	3: while flag[0] and turn=0 do wait
⟨Critical Section⟩:	⟨Critical Section⟩:
⟨Exit⟩:	⟨Exit⟩:
4: flag[0]:=false	4: flag[1]:=false
⟨Remainder⟩:	⟨Remainder⟩:

Algorithm 2

Initially flag[0] and flag[1] are false

code for p_0	code for p_1
⟨Entry⟩:	⟨Entry⟩:
1: while flag[1] do wait	1: while flag[0] do wait
2: flag[0]:=true	2: flag[1]:=true
⟨Critical Section⟩:	⟨Critical Section⟩:
⟨Exit⟩:	⟨Exit⟩:
3: flag[0]:=false	3: flag[1]:=false
⟨Remainder⟩:	⟨Remainder⟩:

Algorithm 3

Initially turn is 0

code for p_0	code for p_1
⟨Entry⟩:	⟨Entry⟩:
1: while turn=1 do wait	1: while turn=0 do wait
⟨Critical Section⟩:	⟨Critical Section⟩:
⟨Exit⟩:	⟨Exit⟩:
2: turn:=1	2: turn:=0
⟨Remainder⟩:	⟨Remainder⟩: