Κωδικός Εργασίας (2.3)

**Ομάδα** [2]

**Μέλος A**: [Θεοδοσίου Κωνσταντίνος, 1619, konstheo@uth.gr]

**Μέλος B**: [Καράκουλα Κατερίνα, 1604, karakoul@uth.gr]

## Περιγραφή σημείων συγχρονισμού με ψευτοκώδικα

|  |
| --- |
| (α) Για το ride και το mtx\_pass. To ride το χρησιμοποιουμε για να δωσουμε σημα να επιβιβαστει ο επιβατης στο roller\_coaster. Και το το mtx\_pass το χρησιμοποιουμε για να δωσουμε σημα στη main οτι εγινε η επιβιβαση. |
| (β) main:  init(mtx\_pass), init(ride)  lock(ride), lock(mtx\_pass)  while(1) {  /\*energopoihsh roller\_coaster\*/  while(i<N) {  unlock(ride)  lock(mtx\_pass)  }  }  passenger:  lock(ride)  unlock(mtx\_pass)  /\*dhlwsh epivivashs\*/ |
| (γ) Παρατηρουμε οτι το προγραμμα μας δεν θα λειτουργησει σωστα αν γινει πρωτα το unlock(ride) στη main πριν προλαβει να κανει lock ο passenger, ουσιαστικα να χασουμε unlock. Το ιδιο ισχυει και με το mtx\_pass. |

## Περιγραφή σημείων συγχρονισμού με ψευτοκώδικα

|  |
| --- |
| (α) Για το roller και το mtx\_rc. Το roller ειδοποιει το roller\_coaster οτι ειναι ετοιμο να ξεκινησει καινουρια βολτα και το mtx\_rc οτι μολις τελειωσε τη βολτα του. |
| (β)main:  init(roller), init(mtx\_rc)  lock(roller), lock(mtx\_rc)  while(epivates>N) {  unlock(roller)  /\*eidopoiei N epivates na boun sthn koursa\*/  lock(mtx\_rc)  }  roller\_coaster:  while(1) {  lock(roller)  /\*volta\*/  unlock(mtx\_rc)  } |
| (γ) Το προβλημα ειναι το ιδιο και με το προηγουμενο σημειο συγχρονισμου. Δηλαδη υπαρχει περιπτωση να γινει πρωτα unlock στη main πριν παρει τον επεξεργαστη το roller\_coaster, και παλι χανεται μια unlock. |

## Περιγραφή σημείων συγχρονισμού με ψευτοκώδικα

|  |
| --- |
| (α) Το msg\_pass ειναι για τον αμοιβαιο αποκλεισμο μεταξυ των passenger. Για να αποφευχθει ταυτοχρονη επεξεργασια του κρισιμου κωδικα. |
| (β) main:  init(msg\_pass)  passenger :  lock(msg\_pass)  /\*krisimos kwdikas\*/  unlock(msg\_pass) |

## Περιγραφή σημείων συγχρονισμού με ψευτοκώδικα

|  |
| --- |
| (α) To mtx το χρησιμοποιουμε για τον αμοιβαιο αποκλεισμο αναμεσα στη main και passenger για να παρει ο passenger το σωστο αριθμο “εισητηριου” απο τη main. |
| (β) main:  init(mtx)  lock(mtx)  for(Npass) {  i++;  thread\_create(passenger, “arithmos eishthriou” &i)  lock(mtx)  }  passenger:  k = i (arithmos eishthriou autou tou thread)  unlock(mtx)  /\*perimenei sthn oura gia thn koursa\*/ |
| (γ) Ισχυει το ιδιο με τους προηγουμενους συγχρονισμους.Χανεται unlock σε περιπτωση που παρει τον επεξεργαστη πρωτα ο passenger. |

## Περιγραφή σημείων συγχρονισμού με ψευτοκώδικα

|  |
| --- |
| (α) Ο τελευταιος επιβατης καθε κουρσας δηλωνει με το msg\_rc στο roller\_coaster οτι επιβιβαστηκε και μπορει να ξεκινησει η κουρσα. |
| (β) main:  init(msg\_rc)  lock(msg\_rc)  roller\_coaster:  while(1) {  lock(msg\_rc)  /\*ekteleitai h koursa\*/  }  passenger:  /\*epivivash passenger\*/  if(passenger == N) {  unlock(msg\_rc)  } |
| (γ) Υπαρχει το ιδιο προβλημα με τα προηγουμενα σημεια συγχρονισμου. Δηλαδη να γινει πρωτα το unlock στον passenger. |