Κωδικός Εργασίας (3.3)

**Ομάδα** [2]

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

**Μέλος B**: [Κωνσταντίνος Θεοδοσίου, 1619, [konstheo@uth.gr](mailto:konstheo@uth.gr)]

## Περιγραφή monitor

|  |
| --- |
| (α) Υλοποιουμε το monitor μας με eggshel signal and continue. Για να το επιτυχουμε αυτο οταν ο χρηστης καλει synch\_notify/notifyall εχουμε εναν μετρητη ο οποιος αυξανεται εφοσον εχει προηγηθει synch\_wait(). Ομως για να υποστηριξουμε τη signal and continue το unlock της εσωτερικης ουρας ειτε στη synch\_end ειτε στη synch\_wait. |

## Περιγραφή σημείων συγχρονισμού με ψευτοκώδικα

|  |
| --- |
| (α) Για τη συναρτηση synch, η οποια δηλωνει ως global και αρχικοποιει τις μεταβλητες του monitor μας. Αρχικοποιουμε το cond\_label και το μπλοκαρουμε ωστε το επομενο που θα μπει στην εσωτερικη ουρα cond\_label να μπλοκαρει. |
| (β) synch(label) {  int q\_label, flag\_label;  pthread\_mutex\_t mtx\_label;  pthread\_mutex\_t cond\_label;    q\_label=0;  flag\_label=0;  init(mtx\_label);  init(cond\_label);  lock(cond\_label);  } |

## Περιγραφή σημείων συγχρονισμού με ψευτοκώδικα

|  |
| --- |
| (α) Για τη συναρτηση synch\_begin, η οποια ενεργοποιει το πλαισιο συγχρονισμου του label. Πιο συγκεκριμενα, οσα threads καλεσουν synch\_begin(label) μπαινουν στην ουρα αναμονης για να χρησιμοποιησουν το monitor με το ιδιο label. |
| (β) synch\_begin(label) {  lock(mtx\_label);  } |

## Περιγραφή σημείων συγχρονισμού με ψευτοκώδικα

|  |
| --- |
| (α) Για τη συναρτηση synch\_notify, η οποια μολις καλεστει απο καποιο thread αυξανει τη μεταβλητη flag\_label(το αναγνωριστικο label ειναι αυτο που εχει δηλωθει τελευταιο στην εναρξη του monitor), εφοσον εχει προηγηθει wait και εχει αρχισει το πλαισιο συγχρονισμου. |
| (β) synch\_notify() {  if(q\_label>0) {  flag\_label ++;  }  } |

## Περιγραφή σημείων συγχρονισμού με ψευτοκώδικα

|  |
| --- |
| (α) Για τη συναρτηση synch\_notifyall, η οποια μολις καλεστει απο καποιο thread κανει τη μεταβλητη flag\_label= q\_label(το αναγνωριστικο label ειναι αυτο που εχει δηλωθει τελευταιο στην εναρξη του monitor) , εφοσον εχει προηγηθει wait και εχει αρχισει το πλαισιο συγχρονισμου. |
| (β) synch\_notifyall() {  if(label>0) {  flag\_label = q\_label;  }  } |

## Περιγραφή σημείων συγχρονισμού με ψευτοκώδικα

|  |
| --- |
| (α) Για τη συναρτηση synch\_wait, η οποια ελεγχει αν εχει γινει notify/notifyall και ενεργοποιει το/τα thread(s) που βρισκεται στην εσωτερικη ουρα και μετα μπλοκαρει στο τελος της ιδιας ουρας. |
| (β) synch\_wait() {  if(flag\_label>0){  flag\_label--;  q\_label--;  unlock(cond\_label);  }  else {  unlock(mtx\_label);  }    q\_label++;  lock(cond\_label);  } |
| (γ) Το προβλημα που υπαρχει ειναι αν καποιο αλλο thread προλαβει και μπει στο monitor και κανει wait και μπαινει στην ουρα cond\_label πριν προλαβει το προηγουμενο να παρει τη θεση του στην ουρα cond\_label.Δηλαδη υπαρχει πιθανοτητα προσπερασης. |

## Περιγραφή σημείων συγχρονισμού με ψευτοκώδικα

|  |
| --- |
| (α) Για τη συναρτηση synch\_end, η οποια ελεγχει αν εχει γινει notify/notifyall και ενεργοποιει το/τα thread(s) που βρισκεται στην εσωτερικη ουρα. Αλλιως επιτρεπει να μπει οποιοσδηποτε αλλος στο monitor. |
| (β) synch\_end(label) {  if(flag\_label>0){  flag\_label--;  q\_label--;  pthread\_mutex\_unlock(&cond\_label);  }  else{  pthread\_mutex\_unlock(&mtx\_label);  }  } |