

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΠΑ 222 — ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ (7.5 ECTS)

Ακαδημαϊκό Έτος 2016-2017, 4ο Εξάμηνο

Εξέταση Ημιεξαμήνου

Ημερομηνία : 19 Μαρτίου 2017
Διάρκεια εξέτασης : 2:30 ώρες
Διδάσκων καθηγητής : Γιώργος Α. Παπαδόπουλος

Απαντήστε όλες τις ερωτήσεις. Ο αριθμός των μονάδων της κάθε (υπο-) ερώτησης φαίνεται σε παρένθεση.

1. α) Θεωρείστε δύο διεργασίες:

```
process A(...)                process B(...)
{
  while (1)
  {
    statement 1;
    statement 2;
  }
}
parbegin A(); B(); parend
```

Το ζητούμενο είναι ότι το statement 1 θα πρέπει να εκτελεστεί πριν από το statement 4 και το statement 3 θα πρέπει να εκτελεστεί πριν από το statement 2. Δεν μας ενδιαφέρει η σειρά με την οποία θα εκτελεστούν το statement 1 σε σχέση με το statement 3 ή το statement 2 σε σχέση με το statement 4. Κάνοντας χρήση γενικών σημαφόρων, υλοποιήστε το ζητούμενο αυτό, με τον πιο απλό δυνατό τρόπο. (10%)

β) Γενικεύοντας το ανωτέρω σενάριο, θεωρείστε μία ομάδα από διεργασίες:

```
process A(...)    process B(...)    ...    process Z(...)
{
  .....
  <meeting point> <meeting point>    <meeting point>
  .....
}
parbegin A(); B(); ... Z(); parend
```

Το ζητούμενο εδώ είναι ότι πριν μία διεργασία αρχίσει να εκτελεί τον κώδικα που βρίσκεται κάτω από τη γραμμή <meeting point>, θα πρέπει πρώτα όλες οι διεργασίες να έχουν ολοκληρώσει την εκτέλεση του κώδικα που βρίσκεται πάνω από τη γραμμή <meeting point>. Σημειωτέον, ότι ο κώδικας των διεργασιών δεν περιλαμβάνει βρόχους. Για αυτό το σενάριο, έχει δοθεί η ακόλουθη λύση:

```
int n=N,          /* number of processes */
    count=0;
sema sync=0;

process A(...)    process B(...)    ...    process Z(...)
{
  .....
  count++;
  if (count==n)
  .....
  count++;
  if (count==n)
  .....
  count++;
  if (count==n)
```

```

    signal(sync);          signal(sync);          signal(sync);
    wait(sync);            wait(sync);            wait(sync);
    <meeting point>      <meeting point>      <meeting point>
    .....
}                          }                          }
parbegin A(); B(); ... Z(); parend

```

Η εξήγηση της λύσης είναι η εξής: Οι πρώτες $N-1$ διεργασίες που φτάνουν στο σημείο <meeting point> τίθενται υπό αναστολή στο σημαφόρο sync περιμένοντας τη διεργασία N να φτάσει και αυτή στο ίδιο σημείο για να μπορέσουν όλες να συνεχίσουν την εκτέλεση του κώδικα κάτω από το σημείο <meeting point>.

(i) Εξηγήστε γιατί η ανωτέρω λύση είναι λανθασμένη και διορθώστε την. (5%)

(ii) Αν η εκτέλεση του κώδικα των διεργασιών βρισκόταν μέσα σε βρόχο, εξηγήστε γιατί η λογική της ανωτέρω λύσης είναι λανθασμένη (δεν χρειάζεται να δώσετε τη σωστή λύση). (5%)

2. Σε μία αίθουσα χορού, καβαλιέροι και ντάμες περιφέρονται συζητώντας. Αν ένας καβαλιέρος θελήσει να χορέψει, ψάχνει να βρει μία ντάμα που να περιμένει κάποιον καβαλιέρο. Αν υπάρχει ντάμα, τότε ξεκινούν να χορεύουν, διαφορετικά περιμένει. Παρομοίως, αν μία ντάμα θέλει να χορέψει, ψάχνει να βρει έναν καβαλιέρο που περιμένει να χορέψει και ανάλογα με την περίπτωση ξεκινάει να χορεύει μαζί του ή περιμένει. Η βασική δομή του κώδικα είναι η εξής:

```

void leader()                                void queen()
{
    while (1)
    {
        .....
        dance();
        .....
    }
}
parbegin leader(); queen(); parend

```

όπου είναι οι εντολές που υλοποιούν το σενάριο αυτό με χρήση γενικών σημαφόρων. Συμπληρώστε τον κώδικα με τις εντολές αυτές και την εισαγωγή των αναγκαίων σημαφόρων. (15%)

3. Θεωρείστε το ακόλουθο σενάριο που λαμβάνει χώρα σε μία στάση λεωφορείου: Οι επιβάτες περιμένουν το λεωφορείο στη στάση και μόλις αυτό φτάσει αρχίζουν να επιβιβάζονται. Αν κατά τη διάρκεια της επιβίβασης εμφανιστούν και άλλοι επιβάτες, αυτοί οι τελευταίοι πρέπει να περιμένουν το επόμενο λεωφορείο. Η χωρητικότητα του λεωφορείου είναι 50 επιβάτες και αν στη στάση υπάρχουν περισσότεροι από τόσους, οι υπόλοιποι πρέπει να περιμένουν το επόμενο λεωφορείο. Στο σενάριο εμπλέκεται μία διεργασία bus και ένας αριθμός διεργασιών passenger. Μία διεργασία τύπου passenger καλεί στον κώδικά της τη συνάρτηση boardbus() για να επιβιβαστεί ο αντίστοιχος επιβάτης. Όταν όλοι οι επιβάτες που περιμένουν στη στάση έχουν επιβιβαστεί, η διεργασία bus καλεί τη συνάρτηση depart() για να αναχωρήσει το λεωφορείο. Σημειωτέον ότι αν το λεωφορείο φτάσει στη στάση σε στιγμή που δεν περιμένει εκεί κανένας επιβάτης, το λεωφορείο πρέπει να αναχωρήσει αμέσως. Υλοποιήστε το σενάριο αυτό με χρήση γενικών σημαφόρων. (15%)

4. Σε ένα μικτό γυμναστήριο γυμνάζονται και άντρες και γυναίκες. Ο καθένας από αυτούς γυμνάζεται για κάποιο διάστημα και μετά κάνει ντους, και αυτή η διαδικασία επαναλαμβάνεται πολλές φορές (για απλοποίηση του σεναρίου

θεωρούμε ότι γίνεται συνέχεια). Δεν υπάρχει περιορισμός για τον αριθμό των ατόμων που μπορεί να είναι ταυτόχρονα στο χώρο που είναι τα ντους αλλά δεν μπορεί να βρίσκονται ταυτόχρονα σε αυτόν τον χώρο αθλητές και από τα δύο φύλλα. Υλοποιήστε το πρόβλημα αυτό με χρήση ενός παρακολουθητή. Το κρίσιμο τμήμα είναι το ντους και υπάρχουν δύο είδη διεργασιών, `male()` και `female()` για άντρες και γυναίκες αντίστοιχα. Ο παρακολουθητής πρέπει να έχει δύο ζευγάρια συναρτήσεων `enter_male()`, `leave_male()` και `enter_female()`, `leave_female()`, οι οποίες εκτελούνται από τις διεργασίες `male()` και `female()` για να μπει στο ντους ή να βγει από αυτό ένας άντρας ή μία γυναίκα αντίστοιχα. Η λύση σας πρέπει να είναι δίκαιη προς τις δύο κατηγορίες διεργασιών με την εξής έννοια: αν ετοιμάζεται να μπει στο ντους ένας άντρας την ώρα που περιμένει γυναίκα να μπει, τότε η γυναίκα έχει προτεραιότητα και αντιστροφή. **(15%)**

5. Σε ένα σύστημα η κάθε διεργασία έχει ένα μοναδικό ακέραιο αριθμό `pid`. Επίσης, υπάρχει ένα αρχείο στο οποίο ανά πάσα χρονική στιγμή μπορούν να έχουν πρόσβαση ένας αριθμός διεργασιών υπό τον εξής όρο: Το άθροισμα όλων των μοναδικών αριθμών των διεργασιών που έχουν πρόσβαση σε κάποια χρονική στιγμή στο αρχείο αυτό, πρέπει να είναι μικρότερο ενός ορίου `MAX`. Γράψτε έναν παρακολουθητή που να υλοποιεί την πρόσβαση στο αρχείο με βάση αυτόν τον όρο. Ο παρακολουθητής θα έχει τις συναρτήσεις `access(pid)` και `release(pid)` με τις οποίες μία διεργασία μπορεί να αποκτήσει πρόσβαση στο αρχείο και να τερματίσει την πρόσβαση σε αυτό αντίστοιχα. **(16%)**
6. Θεωρείστε ένα σύστημα με 5 διεργασίες Δ και 3 είδη πόρων Π . Ο κατωτέρω πίνακας δείχνει για κάθε διεργασία Δ_i την ποσότητα μονάδων που έχει δεσμεύσει από κάθε είδος πόρων Π_j και τη μέγιστη ποσότητα μονάδων που μπορεί να χρειαστεί από κάθε είδος πόρων, καθώς επίσης και τη συνολική ποσότητα μονάδων από κάθε είδος πόρων.

Διεργασία	Ποσότητα πόρων που έχουν δεσμευτεί από κάθε είδος			Μέγιστη ποσότητα πόρων που τυχόν θα χρειαστεί η διεργασία		
	Π_1	Π_2	Π_3	Π_1	Π_2	Π_3
Δ_0	0	1	0	7	5	3
Δ_1	2	0	0	3	2	2
Δ_2	3	0	2	9	0	2
Δ_3	2	1	1	2	2	2
Δ_4	0	0	2	4	3	3

Τρέχουσα ποσότητα διαθέσιμων μονάδων για κάθε είδος πόρων

Π_1	Π_2	Π_3
3	3	2

Με βάση τον πίνακα αυτό και κάνοντας χρήση της λογικής του αλγόριθμου του τραπεζίτη, απαντήστε τις ακόλουθες ερωτήσεις:

- α) Ποια είναι η συνολική ποσότητα πόρων στο σύστημα; **(1%)**
- β) Πόσες μονάδες από κάθε κατηγορία πόρων είναι δυνατόν να ζητήσει η κάθε διεργασία στο μέλλον; **(1%)**
- γ) Είναι η τρέχουσα κατάσταση ασφαλής και γιατί; **(5%)**
- δ) Μπορούν να δοθούν στη διεργασία Δ_1 1 μονάδα από τον πόρο Π_1 και 2 μονάδες από τον πόρο Π_3 και γιατί; **(5%)**
7. Θεωρείστε ένα σύστημα με 3 διεργασίες Δ και 4 είδη πόρων Π . Ο ακόλουθος πίνακας δείχνει για κάθε διεργασία Δ_i την ποσότητα μονάδων που έχει δεσμεύσει από κάθε είδος πόρων Π_j , την ποσότητα μονάδων που ζητεί να δεσμεύσει, καθώς επίσης και την ποσότητα μονάδων που είναι διαθέσιμες από κάθε είδος πόρων.

Διεργασία	Ποσότητα πόρων που έχουν δεσμευτεί από κάθε είδος				Ποσότητα πόρων που ζητεί να δεσμεύσει η διεργασία			
	$\underline{\Pi 1}$	$\underline{\Pi 2}$	$\underline{\Pi 3}$	$\underline{\Pi 4}$	$\underline{\Pi 1}$	$\underline{\Pi 2}$	$\underline{\Pi 3}$	$\underline{\Pi 4}$
Δ1	0	0	1	0	2	0	0	1
Δ2	2	0	0	1	1	0	1	0
Δ3	0	1	2	0	2	1	0	0
<u>Διαθέσιμη ποσότητα μονάδων για κάθε είδος πόρων</u>								
	$\underline{\frac{\Pi 1}{2}}$	$\underline{\frac{\Pi 2}{1}}$	$\underline{\frac{\Pi 3}{0}}$	$\underline{\frac{\Pi 4}{0}}$				

Με βάση τον πίνακα αυτό δείξτε αν το σύστημα έχει περιέλθει ή όχι σε αδιέξοδο.
(7%)

Σημείωση: Στις απαντήσεις σας πρέπει να φαίνονται καθαρά οι υπολογισμοί που κάνατε για να καταλήξετε σε αυτές. Απλή αναφορά σε αποτελέσματα δεν θεωρείται απάντηση.

Καλή Επιτυχία!