

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΠΑ 222 — ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ (7.5 ECTS)

Ακαδημαϊκό Έτος 2019-2020, 4ο Εξάμηνο

Τελική Εξέταση

Ημερομηνία : 21 Μαΐου 2020
Διάρκεια εξέτασης : 3:00 ώρες
Διδάσκων καθηγητής : Γιώργος Α. Παπαδόπουλος

Απαντήστε ΟΛΕΣ τις ερωτήσεις. Ο βαθμός της κάθε (υπο-) ερώτησης αναφέρεται σε παρένθεση. (Η εξέταση αυτή έγινε διαδικτυακά).

1. Σε ένα δωμάτιο υπάρχει μια ομάδα από ρομπότ, μια σκούπα, και ένας ασύρματος φορτιστής. Το κάθε ρομπότ ή κοιμάται ή σκουπίζει, αφού όμως πρώτα φορτιστεί! Προφανώς μόνο ένα ρομπότ μπορεί ανά πάσα στιγμή να έχει τη σκούπα, αλλά ο φορτιστής μπορεί να υποστηρίξει ταυτόχρονα 3 ρομπότ. Γράψτε κώδικα στη γλώσσα C που μοντελοποιεί το πιο πάνω σενάριο με χρήση σημαφόρων. Θεωρήστε ότι υπάρχουν έτοιμες οι συναρτήσεις `snooze()` – αποκοιμάται , `broom()` – σκουπίζει και `charge()` – φορτίζει. (20 μονάδες)
2. Γράψτε έναν παρακολουθητή στη γλώσσα Java που να υλοποιεί το κατωτέρω σενάριο χρήσης τριών εκτυπωτών που βρίσκονται σε ένα δωμάτιο. Ο παρακολουθητής αυτός πρέπει να έχει δύο μεθόδους, `getPrinter()` και `releasePrinter()` και οι κανόνες χρήσης των εκτυπωτών έχουν ως εξής:
 - (i) Αν μπει ένα χρήστης στο δωμάτιο και υπάρχει ελεύθερος εκτυπωτής, τότε μπορεί να τον χρησιμοποιήσει διαφορετικά περιμένει μέχρι να ελευθερωθεί κάποιος.
 - (ii) Αν ελευθερωθεί κάποιος εκτυπωτής και περιμένει κόσμος, τότε αυτός που περιμένει περισσότερο δικαιούται να το χρησιμοποιήσει.Μπορείτε να χρησιμοποιήσετε την κατηγορία `Queue` της Java αν θέλετε. Θεωρήστε ότι στον κώδικα του χρήστη υπάρχει ήδη συμπληρωμένη η μέθοδος `usePrinter()`. Δεν χρειάζεται να δώσετε τον κώδικα για τα `import statements` αλλά πρέπει να δώσετε μια `main` καθώς και τον κώδικα του χρήστη. (20 μονάδες)
3. Ακολουθεί ο σκελετός ενός προγράμματος C:

```
#include <stdio.h>
#include <semaphore.h>
... semaphore declarations ...
void A(void* a){
... wait/signal operations ...
}
```

```

void B(void* b){
... wait/signal operations ...
}

int main() {
pthread_t A1, A2, A3, B1;
... semaphore initialization ...
pthread_create(&A1, NULL, (void*)A, NULL);
pthread_create(&A2, NULL, (void*)A, NULL);
pthread_create(&A3, NULL, (void*)A, NULL);
pthread_create(&B1, NULL, (void*)B, NULL);

pthread_join(A1, NULL);
pthread_join(A2, NULL);
pthread_join(A3, NULL);
pthread_join(B1, NULL);
}

```

Συμπληρώστε τον κώδικα (τα έντονα μαυρισμένα κομμάτια) έτσι ώστε ο τερματισμός εκτέλεσης των συναρτήσεων να είναι ΠΑΝΤΑ με την εξής σειρά: A (οποιοδήποτε αντίγραφο), B, A, A. (10 μονάδες)

4. Σε ένα σύστημα υπάρχουν 4 διεργασίες, Δ1-Δ4, και 2 κατηγορίες πόρων Π1-Π2, όπου κάθε κατηγορία πόρων έχει 2 πόρους. Επίσης ισχύουν οι εξής σχέσεις μεταξύ των διεργασιών και των πόρων:
- Η Δ1 έχει δεσμεύσει 1 πόρο από την κατηγορία Π2 και ζητά 1 πόρο από την κατηγορία Π1.
 - Η Δ2 έχει δεσμεύσει 1 πόρο από την κατηγορία Π1 και δεν χρειάζεται άλλους πόρους.
 - Η Δ3 έχει δεσμεύσει 1 πόρο από την κατηγορία Π1 και ζητά 1 πόρο από την κατηγορία Π2.
 - Η Δ4 έχει δεσμεύσει 1 πόρο από την κατηγορία Π2 και δεν χρειάζεται άλλους πόρους.
- (i) Δημιουργήστε τον γράφο εκχώρησης πόρων για το ανωτέρω σενάριο. (3 μονάδες)
- (ii) Υπάρχουν κυκλικές σχέσεις στον γράφο; Αν ναι, αναφέρατε ποιες είναι. (3 μονάδες)
- (iii) Βρίσκεται το σύστημα σε αδιέξοδο; Αν ναι, εξηγήστε γιατί. Αν όχι, δώστε μια πιθανή ακολουθία εκτέλεσης των διεργασιών, έτσι ώστε όλες οι διεργασίες να ολοκληρώσουν την εκτέλεση τους. (3 μονάδες)
5. Θεωρείστε τέσσερις μαθηματικούς, Δ1, Δ2, Δ3 και Δ4, οι οποίοι κάθονται γύρω από ένα τραπέζι. Μεταξύ των μαθηματικών που κάθονται δίπλα δίπλα, υπάρχει ένα πιρούνι (σύνολο 4 πιρούνια στο τραπέζι, Α, Β, Γ και Δ). Ο κάθε μαθηματικός είτε σκέπτεται είτε τρώει. Για να φάει, χρειάζεται τα πιρούνια δεξιά και αριστερά του. Π.χ. ο μαθηματικός Δ1 χρειάζεται τα πιρούνια Α και Β. Η κατάσταση αυτή σχετίζεται και με τους κατωτέρω πίνακες:

Διαθέσιμη ποσότητα πιρουνιών

<u>A</u>	<u>B</u>	<u>Γ</u>	<u>Δ</u>
1	1	1	1

Μαθηματικός Τρέχουσα δέσμευση πιρουνιών Μέγιστη δέσμευση πιρουνιών

	<u>A</u>	<u>B</u>	<u>Γ</u>	<u>Δ</u>	<u>A</u>	<u>B</u>	<u>Γ</u>	<u>Δ</u>
Δ1
Δ2
Δ3
Δ4

- (i) Συμπληρώστε τη δεξιά πλευρά του ανωτέρω πίνακα (κάτω από την επικεφαλίδα “μέγιστη δέσμευση πιρουνιών”), δηλαδή για κάθε μαθηματικό δηλώστε τα πιρουνία που θα χρειαστεί για να μπορεί να φάει. (3 μονάδες)
- (ii) Συμπληρώστε την αριστερή πλευρά του ανωτέρω πίνακα (κάτω από την επικεφαλίδα “τρέχουσα δέσμευση πιρουνιών”), έτσι ώστε να απεικονίζει μια ΜΗ ΑΣΦΑΛΗ κατάσταση. Επιπλέον, δείξτε μια σειρά από αιτήσεις δέσμευσης πιρουνιών η οποία οδηγεί στη μη ασφαλή κατάσταση που απεικονίζει η αριστερή πλευρά του πίνακα. Η κάθε αίτηση πρέπει να έχει τη μορφή “Ο δικηγόρος X ζητά να δεσμεύσει το πιρούνι Y”. (3 μονάδες)
- (iii) Θεωρείστε τη σειρά από αιτήσεις στην υποερώτηση (ii) που οδήγησαν σε μη ασφαλή κατάσταση και τώρα χρησιμοποιείτε τον αλγόριθμο του τραπεζίτη για να αποφύγετε να βρεθείτε σε μη ασφαλή κατάσταση. Δείξτε τη σειρά με την οποία θα φάνε οι μαθηματικοί. (3 μονάδες)
- (iv) Ας υποθέσουμε ότι υλοποιούμε τον κάθε μαθηματικό ως μια διεργασία και κάνουμε χρήση κρίσιμων τμημάτων για πρόσβαση σε κάθε πιρούνι. Αντί για την εφαρμογή του αλγόριθμου του τραπεζίτη, όταν διαπιστώσουμε ότι το σύστημα έχει περιέλθει σε αδιέξοδο τερματίζουμε αμέσως μια διεργασία. Θα αντιμετωπιστεί έτσι το αδιέξοδο αποτελεσματικά; (3 μονάδες)

6. Σε ένα Λ.Σ. με μία ΚΜΕ και μία μονάδα E/E καταφθάνουν 3 διεργασίες για εκτέλεση με τις ακόλουθες ιδιότητες:

	P1	P2	P3
Χρόνος άφιξης	0	2	4
Χρήση ΚΜΕ και E/E σε μονάδες χρόνου	ΚΜΕ: 10 E/E: 10	ΚΜΕ: 6 E/E: 5 ΚΜΕ: 2 E/E: 2	ΚΜΕ: 1 E/E: 1 ΚΜΕ: 2 E/E: 3

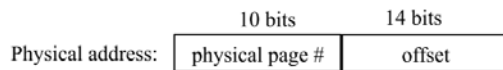
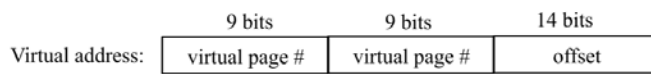
Το Λ.Σ. χρησιμοποιεί για τη χρονοδρομολόγηση των διεργασιών τον αλγόριθμο με βάση τον υπολειπόμενο χρόνο εκτέλεσης (shortest remaining time). Οι αιτήσεις για E/E εξυπηρετούνται με βάση τη σειρά άφιξης (first come first served).

- (i) Αντιγράψτε και συμπληρώστε τον παρακάτω πίνακα εκτέλεσης (η στήλη “ΚΜΕ” δείχνει τη διεργασία που εκτελείται και η στήλη “E/E” δείχνει τη διεργασία που κάνει E/E) (9 μονάδες):

t	Ουρά ΚΜΕ	ΚΜΕ	Ουρά E/E	E/E
0	P1/10	P1	-	-
2	P1/8, P2/6	?	?	?
?				
8	P1/8, P2/3	P2	P3/3	P3
?				
31	-	-	-	-

(ii) Υπολογίστε το ποσοστό χρήσης της ΚΜΕ στο ανωτέρω σενάριο. (4 μονάδες)

7. Σε ένα Λ.Σ. υποστηρίζεται η τεχνική της ιδεατής μνήμης με τις ακόλουθες δομές για ιδεατή μνήμη και φυσική μνήμη:



- (i) Ποιο είναι το μέγεθος μίας σελίδας; (2 μονάδες)
- (ii) Ποιο είναι το συνολικό μέγεθος της φυσικής μνήμης; (2 μονάδες)
- (iii) Αν χρησιμοποιηθεί ανεστραμμένος πίνακας σελίδων, πόσα στοιχεία θα έχει αυτός ο πίνακας; (2 μονάδες)
8. Σε ένα σύστημα σελιδοποίησης, το Λ. Σ. χορηγεί σε μία διεργασία 3 πλαίσια σελίδων. Η διεργασία αυτή αποτελείται από 4 σελίδες και αναφορά σε αυτές γίνεται με την εξής σειρά: A, B, D, C, B, A, B, A. Στην αρχή όλα τα πλαίσια σελίδων είναι άδεια. Δείξτε πως θα γίνεται η φόρτωση και εναλλαγή των σελίδων στην κύρια μνήμη και υπολογίστε τα σφάλματα σελίδων που θα προκύψουν για τους εξής αλγόριθμους αντικατάστασης σελίδων:
- (i) Πρώτη-εισερχόμενη-πρώτη-εξερχόμενη (FIFO). (2 μονάδες)
- (ii) Λιγότερο-πρόσφατα-χρησιμοποιούμενης-σελίδας (LRU). (2 μονάδες)
- (iii) Βέλτιστος (optimal). (2 μονάδες)
9. Θεωρείστε τις ακόλουθες 6 αιτήσεις για πρόσβαση σε ένα δίσκο και τον αντίστοιχο διάυλο του δίσκου στον οποίο αναφέρεται η κάθε αίτηση:

Αίτηση:	A	B	C	D	E	F
Διάυλος:	4	10	35	62	69	95

Αν ο χρόνος αναζήτησης (seek time) για μετακίνηση της κεφαλής του δίσκου από τον ένα διάυλο στον επόμενο είναι 0.1 ms, υπολογίστε τον συνολικό χρόνο αναζήτησης για τους αλγόριθμους:

- (i) SSTF (η κεφαλή του δίσκου βρίσκεται στο διάυλο 33). (2 μονάδες)
- (ii) LOOK (η κεφαλή του δίσκου βρίσκεται στο διάυλο 33 με κατεύθυνση τους χαμηλότερους αριθμούς). (2 μονάδες)