

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΠΑ 222 — ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ (7.5 ECTS)

Ακαδημαϊκό Έτος 2020-2021, 4ο Εξάμηνο

Τελική Εξέταση

Ημερομηνία : 22 Μαΐου 2021
Διάρκεια εξέτασης : 2:30 ώρες
Διδάσκων καθηγητής : Γιώργος Α. Παπαδόπουλος

Απαντήστε ΟΛΕΣ τις ερωτήσεις. Ο βαθμός της κάθε (υπο-) ερώτησης αναφέρεται σε παρένθεση.

1. Σε μια μικρή παραλία, υπάρχει μια βάρκα 5 θέσεων που μεταφέρει παραθεριστές από τη ακτή πίσω σε ένα εκδρομικό γιοτ, αλλά όχι αντίστροφα (πάνε κολυμπώντας μέχρι την ακτή). Η βάρκα επιστρέφει στο γιοτ μόνο όταν γεμίσει και αφού επιβιβαστούν στο γιοτ επιστρέφει στην ακτή για να πάρει τους επόμενους, κ.ο.κ. Θεωρείστε ότι υπάρχει πάντα ένας άπειρος αριθμός από παραθεριστές που περιμένουν να εξυπηρετηθούν. Χρησιμοποιώντας σημαφόρους δώστε ένα πρόγραμμα σε C, για την υλοποίηση του ανωτέρω σεναρίου.

ΠΡΟΣΟΧΗ: Συμπληρώστε την απάντησή σας στο αντίστοιχο φύλλο για αυτή την άσκηση στο τετράδιο απαντήσεων! (18%)

2. Σε ένα υποβρύχιο υπάρχουν N ναύτες οι οποίοι μοιράζονται μεταξύ τους τα κρεβάτια του υποβρυχίου (υπάρχουν $2/3 N$ κρεβάτια). Ένας ναύτης κοιμάται για λίγο και μετά μαγειρεύει (το υποβρύχιο είναι στο λιμάνι και δεν βγάζουν βάρδιες). Για να μαγειρέψει ένας ναύτης χρειάζεται καπέλο και ποδιά του σεφ τα οποία φοριούνται αναγκαστικά ξεχωριστά (πρώτα η ποδιά και μετά το καπέλο). Υπάρχουν δυο ποδιές αλλά μετά από ένα ατύχημα έχει μείνει μόνο ένα καπέλο. Αφού μαγειρέψει επιστρέφει το καπέλο και την ποδιά. Προφανώς για να κοιμηθεί ένας ναύτης πρέπει να υπάρχει διαθέσιμο κρεβάτι. Αν κάποιος από αυτούς τους πόρους δεν είναι διαθέσιμος (κρεβάτι για να κοιμηθεί ή καπέλο και ποδιά για να μαγειρέψει) ο ναύτης πρέπει να περιμένει (αλλά αν περιμένει πολύ ώρα θα αρχίσει να νευριάζει).

Γράψτε έναν παρακολουθητή σε Java που να υλοποιεί το ανωτέρω σενάριο. Βεβαιωθείτε ότι η λύση σας δεν υποφέρει από προβλήματα αδιέξοδου ή παρατεταμένης στέρησης. Εκτός από τις μεθόδους του παρακολουθητή, δώστε και 2 μεθόδους που υλοποιούν τον ύπνο και το μαγείρεμα (δώστε μόνο την υπογραφή τους χωρίς πραγματικό κώδικα υλοποίησης). Τέλος, σε ξεχωριστή κατηγορία δώστε και μια `main` που δημιουργεί N ($N \geq 6$ και πολλαπλάσιο του 3) ναύτες (νήματα) που χρησιμοποιούν τον παρακολουθητή σας.

ΠΡΟΣΟΧΗ: Συμπληρώστε την απάντησή σας στο αντίστοιχο φύλλο για αυτή την άσκηση στο τετράδιο απαντήσεων! (20%)

3. Οι δυο ομάδες του Survivor (κόκκινοι και μπλε) πρέπει να πάνε σε ένα νησάκι για μια δοκιμασία. Έχουν μείνει από 8 παίκτες στη κάθε ομάδα και έχουν μια βάρκα με 4 θέσεις για να πάνε στο νησί, η οποία βάρκα όμως πρέπει να είναι πάντα γεμάτη για να χρησιμοποιηθεί, (χρειάζονται 4 κωπηλάτες). Για να μη πλακωθούν καθοδόν, δεν επιτρέπεται να βρίσκονται μαζί στη βάρκα ένας παίχτης από την μία ομάδα και τρεις

από την άλλη – όλοι οι άλλοι συνδυασμοί επιτρέπονται. Συμπληρώστε τον ακόλουθο σκελετό σε C με **σημαφόρους** (μόνο τα έντονα μαυρισμένα κομμάτια) για να επιλύσετε το ανωτέρω πρόβλημα ταυτοχρονίας. Η λύση σας δεν πρέπει να υποφέρει από προβλήματα αδιέξοδου (αλλά δε χρειάζεται να ανησυχήσετε για παρατεταμένη στέρηση). Από τη στιγμή που υπάρχει ένας επιτρεπτός συνδυασμός παιχτών, η μεταφορά τους θα πρέπει να ξεκινήσει αμέσως). Θεωρήστε ότι οι πράξεις +=, -= καθώς και οι συγκρίσεις γίνονται ατομικά (δεν μεσολαβεί εναλλαγή νημάτων κατά την εκτέλεση τους).

ΠΡΟΣΟΧΗ: Συμπληρώστε την απάντηση σας στο αντίστοιχο φύλλο για αυτή την άσκηση στο τετράδιο απαντήσεων! (12%)

```
#include <stdio.h>
#include <semaphore.h>

int redWait = 0, blueWait = 0;
// Declare your semaphores

void rowboat() {
    while (1) {
        //...wait/signal operations...

        if (redWait>1) {
            redWait-=2;
            //...wait/signal ops...
        } else if (blueWait>1) {
            blueWait-=2;
            //...wait/signal ops...
        }

        if (blueWait>1) {
            blueWait-=2;
            //...wait/signal ops...
        } else if (redWait>1) {
            redWait-=2;
            //...wait/signal ops...
        }
    }
}

void redArrives(void *a){
    while (1) {
        //...wait/signal operations...
        redWait+=2;
        //...wait/signal operations...
    }
}

void blueArrives(void *a){
    while (1) {
        //...wait/signal operations...
        blueWait+=2;
        //...wait/signal operations...
    }
}

void redP(void *r){
    //...wait/signal operations...
}

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

int main() {
    int i;
    pthread_t R, B, P[16];
    //... semaphore initialization ...

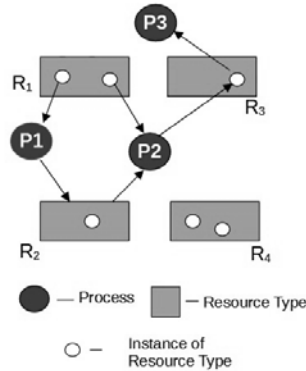
    pthread_create(&R, NULL, (void*)redArrives, NULL);
    pthread_create(&B, NULL, (void*)blueArrives, NULL);

    for(i=0;i<16;i+=2) {
        pthread_create(&P[i], NULL, (void*)redP, NULL);
        pthread_create(&P[i+1], NULL, (void*)blueP, NULL);
    }

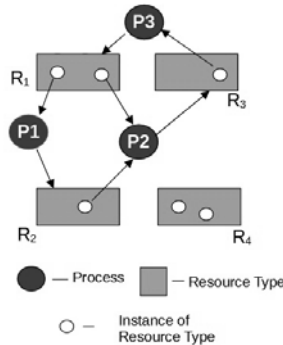
    rowboat();
}
```

4. Σε ένα σύστημα υπάρχουν 3 διεργασίες, P1-P3, και 4 κατηγορίες πόρων R1-R4, όπου κάθε κατηγορία πόρων έχει τον αριθμό των πόρων που φαίνεται στο κάθε σχήμα κατωτέρω.

(i) Αναφορικά με τον ακόλουθο γράφο εκχώρησης πόρων, δηλώστε αν το σύστημα βρίσκεται 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. (1%)



(ii) Επαναλάβετε την υποερώτηση (i) και για τον ακόλουθο γράφο. (1%)



(iii) Θεωρείστε ένα σύστημα με 4 πόρους της ίδιας κατηγορίας οι οποίοι διαμοιράζονται μεταξύ 3 διεργασιών, καμιά από τις οποίες δεν θα χρειασθεί περισσότερο από 2 πόρους. Επιχειρηματολογήστε γιατί δεν μπορεί να προκληθεί αδιέξοδο. (2%)

(iv) Γενικεύοντας το σενάριο της (iii), θεωρείστε ένα σύστημα με M πόρους της ίδιας κατηγορίας οι οποίοι διαμοιράζονται μεταξύ N διεργασιών. Αδιέξοδο δεν μπορεί να υπάρξει αν: (a) Το άθροισμα των μέγιστων αναγκών όλων των διεργασιών είναι $< M+N$. (b) Το άθροισμα των μέγιστων αναγκών όλων των διεργασιών είναι $> M+N$. (c) Ισχύουν και οι δύο συνθήκες a και b. (d) Δεν ισχύει καμιά από τις συνθήκες a ή b. (3%)

(v) Θεωρείστε ένα σύστημα στο οποίο υπάρχει μία κατηγορία πόρων με 5 συνολικά πόρους. Υπάρχουν επίσης 4 διεργασίες, για τις οποίες ισχύουν τα εξής:

<u>Διεργασία</u>	<u>Τρέχουσα δέσμευση πόρων</u>	<u>Μέγιστη δέσμευση πόρων</u>
Δ1	1	2
Δ2	1	3
Δ3	2	4
Δ4	0	5

Επιχειρηματολογήστε για το αν το σύστημα βρίσκεται σε ασφαλή κατάσταση ή όχι και γιατί. (3%)

5. α) Ποια πολιτική χρονοδρομολόγησης διεργασιών θα χρησιμοποιούσατε για κάθε μία από τις ακόλουθες περιπτώσεις και γιατί:
- (i) Οι διεργασίες εμφανίζονται για εκτέλεση με μεγάλα χρονικά διαστήματα μεταξύ τους. (1%)
 - (ii) Ο σημαντικός παράγων είναι το ποσοστό των διεργασιών που ολοκληρώνουν την εκτέλεση τους. (1%)
 - (iii) Όλες οι διεργασίες χρειάζονται λίγο πολύ τον ίδιο χρόνο χρήσης της ΚΜΕ. (1%)
- β) Σε ένα σύστημα οι διεργασίες εμφανίζονται για εκτέλεση με μέσο ρυθμό 6 διεργασίες ανά λεπτό και η κάθε διεργασία χρειάζεται κατά μέσο όρο 8 δευτερόλεπτα χρήσης της ΚΜΕ. Σε τι ποσοστό γίνεται χρήση της ΚΜΕ; (2%)
- γ) Σε ένα σύστημα που χρησιμοποιεί την πολιτική χρονοδρομολόγησης FCFS καταφθάνουν την ίδια χρονική στιγμή 5 διεργασίες με την εξής σειρά σε υπολογιζόμενο χρόνο χρήσης της ΚΜΕ:
- | <u>Διεργασία</u> | <u>Υπολογιζόμενος χρόνος χρήσης ΚΜΕ</u> |
|------------------|---|
| Δ0 | 80 |
| Δ1 | 20 |
| Δ2 | 10 |
| Δ3 | 20 |
| Δ4 | 50 |
- (i) Υπολογίστε τον χρόνο διεκπεραίωσης (turnaround time) για την διεργασία Δ3. (2%)
 - (ii) Υπολογίστε τον μέσο χρόνο αναμονής (mean waiting time) για το ανωτέρω σενάριο. (2%)
- δ) Επαναλάβετε τη (γ) όπου η πολιτική χρονοδρομολόγησης που εφαρμόζεται είναι τώρα η SPN και για την (i) ο υπολογισμός να γίνει για τη διεργασία Δ4. (4%)
6. α) Θεωρείστε μία εφαρμογή αναπαραγωγής βίντεο (video player) όπου υποστηρίζεται η λειτουργικότητα του κανονικού παιχνιδιού αλλά και της δυνατότητας να παιχθεί το βίντεο γρήγορα προς τα μπροστά ή πίσω. Τι πολιτικές διαχείρισης της κύριας μνήμης θα θεωρούσατε ότι είναι κατάλληλες για μία τέτοια εφαρμογή; (3%)
- β) Σε ένα σύστημα που υποστηρίζει σελιδοποίηση, το ποσοστό επιτυχίας εύρεσης μίας σελίδας στην πολύ γρήγορη ενδιάμεση μνήμη μετάφρασης (TLB) είναι 80%. Επίσης η εύρεση μίας σελίδας στην TLB παίρνει 15 μονάδες χρόνου ενώ η εύρεση μίας σελίδας στη συμβατική κύρια μνήμη παίρνει 150 μονάδες χρόνου. Υπολογίστε κατά μέσο όρο πόσο χρόνο παίρνει η πρόσβαση σε κάποια σελίδα. (3%)
- γ) Σε ένα σύστημα με 256 MB κύρια μνήμη και μέγεθος σελίδας 4 KB, πόσα στοιχεία θα έχει ο ανεστραμμένος πίνακας σελίδων; (2%)
- δ) Θεωρείστε ένα λογικό χώρο διευθύνσεων αποτελούμενο από 8 σελίδες, όπου το μέγεθος της σελίδας είναι 1024 bytes, και απεικονιζόμενο στη φυσική μνήμη η οποία αποτελείται από 32 πλαίσια σελίδας.
- (i) Πόσα bits χρειάζονται για να απεικονίσουν τις λογικές διευθύνσεις μνήμης; (1%)
 - (ii) Πόσα bits χρειάζονται για να απεικονίσουν τις φυσικές διευθύνσεις μνήμης; (1%)
- ε) Σε ένα σύστημα που υποστηρίζει σελιδοποίηση, μία διεργασία αποτελείται από 7 σελίδες (A-G) και της έχουν χορηγηθεί 4 πλαίσια. Η πρόσβαση στα περιεχόμενα των σελίδων της διεργασίας γίνεται με την εξής σειρά: A, B, C, D, E, F, C, A, A, F, F, G, A, B, G, D, F, F. Υπολογίστε τον αριθμό των σφαλμάτων σελίδων που θα δημιουργηθούν για κάθε έναν από τους ακόλουθους αλγόριθμους αντικατάστασης σελίδων: (i) FIFO, (2%) (ii) Βέλτιστος, (2%) (iii) LRU. (2%)

7. **α)** Ένας δίσκος είναι επιρρεπής στο να χαλάνε οι τομείς (sectors) του. Με βάση τις διαφορετικές τεχνικές που υπάρχουν για αποθήκευση των αρχείων σε μπλοκ στο δίσκο, προτείνετε μία πολιτική αποθήκευσης όπου σε περίπτωση που χαλάσει ένας τομέας στον οποίο είναι αποθηκευμένο ένα μέρος κάποιου αρχείου, τα υπόλοιπα δεδομένα του αρχείου που είναι αποθηκευμένα σε άλλα μπλοκ παραμένουν προσβάσιμα και δεν χάνονται. **(3%)**
- β)** Ένα αρχείο αποθηκεύεται στο δίσκο με την τεχνική των μπλοκ εγγραφών σταθερού μεγέθους (fixed blocking). Οι εγγραφές του αρχείου έχουν μέγεθος 10 bytes και το μέγεθος του μπλοκ είναι 25 bytes. Σπαταλάται χώρος μέσα σε κάθε μπλοκ και αν ναι πόσος είναι αυτός; **(2%)**
- γ)** Σε συνέχεια της υποερώτησης (β), ας υποθέσουμε τώρα ότι χρησιμοποιείται εναλλακτικά η τεχνική των μπλοκ εγγραφών μεταβλητού μεγέθους χωρίς δυνατότητα σπασίματος (unspanned blocking). Οι εγγραφές του αρχείου έχουν μέγεθος 10 ή 20 bytes και το μέγεθος του μπλοκ παραμένει 25 bytes. Υπολογίστε την ελάχιστη και μέγιστη σπατάλη χώρου που θα μπορούσε να υπάρχει σε κάθε μπλοκ. **(2%)**
- δ)** Ένας δίσκος αποτελείται από 30 μπλοκ, όπου το κάθε μπλοκ έχει μέγεθος 1024 bytes (οπτικά μπορείτε να τον αναπαραστήσετε ως έναν πίνακα 6x5). Υπάρχουν 3 αρχεία με τις εξής ιδιότητες: Το αρχείο F1 αποτελείται από 11 εγγραφές, όπου κάθε εγγραφή έχει μέγεθος 112 bytes. Το αρχείο F2 αποτελείται από 890 εγγραφές, όπου κάθε εγγραφή έχει μέγεθος 13 bytes. Το αρχείο F3 αποτελείται από μία δυαδική ροή μεγέθους 510 bytes. Σχεδιάστε την οργάνωση των αρχείων αυτών στο δίσκο με βάση την τεχνική κατανομής συνδεδεμένης λίστας με δείκτες (indexed allocation). Μπορείτε να επιλέξετε την περίπτωση που δίνεται ένα μπλοκ κάθε φορά ή ομάδα από μπλοκ. **(4%)**