+346/1/60+

Λειτουργικά Συστήματα
Εξεταστική Ιουνίου 2026

◯0 ◯0 ◯0 ◯0 ◯0 ◯0 ◯0
◯1 ◯1 ◯1 ◯1 ◯1 ◯1 ◯1
◯2 ◯2 ◯2 ◯2 ◯2 ◯2 ◯2
◯3 ◯3 ◯3 ◯3 ◯3 ◯3 ◯3
◯4 ◯4 ◯4 ◯4 ◯4 ◯4 ◯4
◯5 ◯5 ◯5 ◯5 ◯5 ◯5 ◯5
◯6 ◯6 ◯6 ◯6 ◯6 ◯6 ◯6
◯7 ◯7 ◯7 ◯7 ◯7 ◯7 ◯7
◯8 ◯8 ◯8 ◯8 ◯8 ◯8 ◯8
◯9 ◯9 ◯9 ◯9 ◯9 ◯9 ◯9

ΔΙΑΡΚΕΙΑ 2 ΩΡΕΣ

Αριστερά: Συμπληρώστε τα ψηφία του Αριθμού Μητρώου σας (ένα ψηφίο ανά στήλη).

Κάτω: Γράψτε το ονοματεπώνυμό σας και τον Αριθμό Μητρώου σας.

Ονοματεπώνυμο και Α.Μ.
.....
.....

ΠΡΟΣΟΧΗ: Οι απαντήσεις που έχουν πλαίσιο απαιτούν αιτιολόγηση μέσα σε αυτό, και βαθμολογούνται βάσει αυτού. Ακόμα κι αν επιλέξετε τη σωστή απάντηση (μαρκάροντας το αντίστοιχο κυκλάκι) χωρίς, όμως, να δώσετε την ανάλογη αιτιολόγηση, ή χωρίς η αιτιολόγηση αυτή να είναι σωστή, η απάντηση εκλαμβάνεται ως λάθος.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!!

Ερώτηση 1: [1.0] Σε disk scheduling, ο αλγόριθμος Scan & Look (γνωστός και ως Elevator) δεν είναι απόλυτα δίκαιος διότι:

◯ Ο χρόνος προσπέλασης των **ακριανών (εξωτ. & εσωτ.)** κυλίνδρων είναι κάπως μεγαλύτερος
◯ Ο χρόνος προσπέλασης των **εσωτερικών (μικρότερων)** κυλίνδρων είναι κάπως μεγαλύτερος
◯ Ο χρόνος προσπέλασης των **εξωτερικών (μεγαλύτερων)** κυλίνδρων είναι κάπως μεγαλύτερος
◯ Ο χρόνος προσπέλασης των **μεσαίων** κυλίνδρων είναι κάπως μεγαλύτερος

Ερώτηση 2: [1.0] Τα νήματα (threads) μιας διεργασίας (process) μοιράζονται την ίδια μνήμη.

◯ Δεν ισχύει ποτέ
◯ Ισχύει πάντα
◯ Μόνο όταν είναι υλοποιημένα σε επίπεδο χρήστη
◯ Μόνο όταν είναι υλοποιημένα σε επίπεδο πυρήνα
◯ Εξαρτάται από το αν τα threads είναι attached ή detached
◯ Ισχύει μόνο για τα threads που ξεκινούν από την ίδια start routine

Ερώτηση 3: [1.0] Έστω μία σημαφόρος s με αρχική τιμή 2 και δύο διεργασίες Δ1 και Δ2. Οι διεργασίες εκτελούν τις παρακάτω πράξεις με την ακόλουθη σειρά: Δ1: down(s), Δ2: down(s), Δ1: up(s), Δ1:up(s), Δ2:down(s), Δ1:down(s), Δ1:down(s). Στο τέλος τι θα έχει συμβεί;

- ◯ Δεν θα είναι μπλοκαρισμένη καμία διεργασία
- ◯ Θα είναι μπλοκαρισμένες και η Δ1 και η Δ2
- ◯ Θα είναι μπλοκαρισμένη η Δ1
- ◯ Θα είναι μπλοκαρισμένη η Δ2

Ερώτηση 4: [1.0] Θέλουμε να επιβάλλουμε thread synchronization μεταξύ δύο threads (προφανώς του ίδιου process), δηλαδή το thread A να περιμένει κάποια ενέργεια του thread B πριν προχωρήσει. Π.χ., πριν το thread A (consumer) διαβάσει δεδομένα από έναν buffer, να περιμένει να γράψει το thread B (producer) δεδομένα στον buffer αυτόν.

Σημειώστε όλα όσα είναι σωστά, και **εξηγήστε στο πλαίσιο** γιατί πρέπει/δεν πρέπει να χρησιμοποιήσουμε mutex, και γιατί πρέπει/δεν πρέπει να χρησιμοποιήσουμε condition variable.

- ◯ χρειάζεται να χρησιμοποιήσουμε mutex
- ◯ δεν χρειάζεται να χρησιμοποιήσουμε mutex
- ◯ χρειάζεται να χρησιμοποιήσουμε condition variable
- ◯ δεν χρειάζεται να χρησιμοποιήσουμε condition variable

Ερώτηση 5: [1.0] Θεωρήστε ένα Unix filesystem με μέγεθος disk block **1KB** και disk address με μήκος **16 bits**. Κάθε i-node έχει:

- 4 direct pointers
- 2 single indirect pointer
- 1 double indirect pointer
- κανέναν triple indirect pointer

Ποιο είναι το μέγιστο δυνατό μέγεθος αρχείου σε αυτό το filesystem; Να υπολογισθεί σε **Kilobytes (KB)** ως άθροισμα δυνάμεων του δύο, και να σημειωθούν όλες αυτές οι δυνάμεις.

◯ 2^0	◯ 2^3	◯ 2^6	◯ 2^9	◯ 2^{12}	◯ 2^{15}	◯ 2^{18}	◯ 2^{21}
◯ 2^1	◯ 2^4	◯ 2^7	◯ 2^{10}	◯ 2^{13}	◯ 2^{16}	◯ 2^{19}	◯ 2^{22}
◯ 2^2	◯ 2^5	◯ 2^8	◯ 2^{11}	◯ 2^{14}	◯ 2^{17}	◯ 2^{20}	◯ 2^{23}

Αναπτύξτε τον υπολογισμό σας στο πλαίσιο:

Ερώτηση 6: [1.0] Σε σύστημα Linux, είστε σε directory που περιέχει ένα αρχείο κειμένου με όνομα `textfile.txt`, και τρέχετε την παρακάτω εντολή:

```
cat textfile.txt | sort | uniq
```

a) Τι κάνει η εντολή;

b) Εάν τρέξετε την παραπάνω εντολή σε υπολογιστή που έχει 8 GB RAM, αλλά το αρχείο `textfile.txt` έχει μέγεθος 50 GB, τι περιορισμός προκύπτει από τη μνήμη; Θα έχουμε out of memory error και θα διακοπεί η εντολή ή θα τρέξει σωστά, και ***γιατί***;

LEAVE EMPTY $\longrightarrow (\bigcirc\bigcirc\bigcirc\bigcirc\bigcirc\cdot\bigcirc\bigcirc\bigcirc)$

Ερώτηση 7: [1.0] Ένα σύστημα διαθέτει πέντε διεργασίες και πέντε εκχωρήσιμους πόρους. Η τρέχουσα κατανομή και οι μέγιστες ανάγκες σε πόρους είναι οι εξής:

	Κατέχει					Μέγιστο					Διαθέσιμοι				
Διεργασία Α	1	0	1	0	1	2	1	2	2	3	1	1	x	1	2
Διεργασία Β	1	0	1	1	1	2	2	1	1	1					
Διεργασία Γ	2	3	1	2	2	2	4	3	2	2					
Διεργασία Δ	2	2	0	2	0	2	2	3	2	1					
Διεργασία Ε	1	1	1	2	1	1	1	2	3	2					

α) Ποια είναι η μικρότερη τιμή του x για την οποία η παραπάνω κατάσταση είναι ασφαλής;

β) Μέσα στο πλαίσιο, δείξτε μια εφικτή σειρά εκτέλεσης των διεργασιών, για την παραπάνω τιμή του x μαζί με τους διαθέσιμους πόρους μετά από την ολοκλήρωση της κάθε μίας.

LEAVE EMPTY $\longrightarrow$ (◯ ◯ ◯ ◯ ◯ · ◯ ◯ ◯)

Ερώτηση 8: [1.0] Ποια από τα παρακάτω στοιχεία *δεν* αποθηκεύονται στο i-node ενός αρχείου; Υπενθυμίζουμε ότι τα i-nodes τα διαχειρίζεται το file service.

- ◯ Τα δικαιώματα πρόσβασης (rwx)
- ◯ Το directory στο οποίο ανήκει το αρχείο
- ◯ Ο ιδιοκτήτης του αρχείου (file owner)
- ◯ Το όνομα του αρχείου (filename)

Ερώτηση 9: [1.0] Όταν μια διεργασία (process) καλεί το system call `exec()`, ποιες από τις παρακάτω προτάσεις είναι αληθείς;

- ◯ Δημιουργείται μια νέα διεργασία, η οποία ξεκινάει την εκτέλεση του προγράμματος που ορίζει η `exec()`
- ◯ Η ίδια η διεργασία που καλεί την `exec()` εγκαταλείπει το τρέχον πρόγραμμα και ξεκινάει την εκτέλεση του προγράμματος που ορίζει η `exec()`
- ◯ Ο αριθμός των threads *πριν* την `exec()` εξακολουθεί να είναι ο ίδιος και *μετά*.
- ◯ Η ίδια η διεργασία που καλεί την `exec()` δημιουργεί νέο thread το οποίο ξεκινάει την εκτέλεση του προγράμματος που ορίζει η `exec()`
- ◯ Τα file descriptors που ήταν ανοιχτά *πριν* την `exec()` εξακολουθούν να είναι ανοιχτά και *μετά*.
- ◯ Δημιουργείται μια νέα διεργασία, αντίγραφο της αρχικής με μόνη διαφορά στην τιμή του pid.

Ερώτηση 10: [1.0] Έστω ο παρακάτω απλουστευμένος πίνακας σελίδων (page table):

	Page Frame	Reference	Modified	Present
7	100	0	1	0
6	010	1	0	1
5	110	0	1	1
4	010	1	0	1
3	000	0	1	1
2	101	1	1	0
1	011	0	0	1
0	001	1	0	0

Σημειώστε όλα όσα από τα παρακάτω είναι αληθή:

- ◯ Ανάγνωση από τη virtual address `01111101` θα προκαλέσει page fault
- ◯ Εγγραφή στη virtual address `00100101` θα προκαλέσει page fault
- ◯ Η virtual address `10010001` αντιστοιχεί στη physical address `01010001`
- ◯ Η virtual address `11001101` αντιστοιχεί στη physical address `01001101`

Στο πλαίσιο δικαιολογήστε συνοπτικά τις επιλογές σας: