ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Λειτουργικά Συστήματα

Εαρινό Εξάμηνο 2021-2022
Θέματα Εξετάσεων 22 Σεπτεμβρίου 2022
Εξεταστής: Γεώργιος Β. Ξυλωμένος

Οδηγίες:

- Διάρκεια εξέτασης: 120 λεπτά.
- Δεν επιτρέπεται η χρήση σημειώσεων, βιβλίων και κινητών τηλεφώνων.
- Κάθε θέμα αναφέρει το βαθμολογικό του βάρος.
- Μπορείτε να απαντήσετε σε όσα θέματα θέλετε.
- Ο μέγιστος δυνατός βαθμός είναι 70 μονάδες.

Θέματα:

1. [10] Και ο συγχρονισμός διεργασιών με σημαφόρους (semaphores) και ο συγχρονισμός νημάτων με mutex (όπως στα pthreads) τελικά υλοποιούνται με εντολές test and set ή compare and swap, δηλαδή βασίζονται στον έλεγχο κάποιας θέσης της μνήμης. Πού πρέπει να βρίσκεται αυτή η θέση μνήμης στους σηματοφόρους, στον πυρήνα ή στη διεργασία, και γιατί; Πού μας συμφέρει να βρίσκεται στα νήματα, στον πυρήνα ή στην κοινή μνήμη των νημάτων, και γιατί;
2. [10] Εξηγήστε γιατί σε ένα σύστημα που υποστηρίζει πολυνηματικές διεργασίες, μπορούμε να έχουμε την απόδοση των ασύγχρονων κλήσεων E/E (asynchronous I/O calls ή non-blocking calls), χρησιμοποιώντας μόνο σύγχρονες κλήσεις E/E (synchronous I/O calls ή blocking calls).
3. [10] Έστω ένας εξυπηρετητής ιστοσελίδων (web server) ο οποίος χρησιμοποιεί όλη τη διαθέσιμη μνήμη του συστήματος ως κρυφή μνήμη (cache) ιστοσελίδων. Για να εξυπηρετεί πολλούς πελάτες ταυτόχρονα, ο εξυπηρετητής χρησιμοποιεί μία διεργασία ανά αίτημα, με τις διεργασίες να μοιράζονται την κρυφή μνήμη μέσω του μηχανισμού κοινής μνήμης (shared memory). Τι θα κερδίσουμε αν ξαναγράψουμε τον εξυπηρετητή ώστε να χρησιμοποιεί ένα νήμα ανά αίτημα;
4. [10] Μία διεργασία αποτελείται από διάφορα τμήματα μνήμης (κώδικα, στοίβα, σωρό, σταθερές, αρχικοποιημένες μεταβλητές, κατάσταση καταχωρητών). Ποια από αυτά πρέπει να αποθηκεύονται στην περιοχή εναλλαγής (swap area) όταν δεν χωράνε στη μνήμη και γιατί;
5. [10] Γιατί όταν αλλάζει η εκτελούμενη διεργασία (context switch) πρέπει να καθαρίζεται η ενδιάμεση/κρυφή μνήμη υποστήριξης μετάφρασης (TLB); Τι θα συμβεί αν χρησιμοποιήσουμε τα παλιά περιεχόμενά της κατά την εκτέλεση της νέας διεργασίας;
6. [10] Γιατί στα συστήματα με διευθύνσεις 64 bit χρησιμοποιούμε κατακερματισμένους (hashed) ή ανεστραμμένους (inverted) και όχι ιεραρχικούς πίνακες σελίδων (hierarchical page tables);
7. [10] Έστω ότι σε έναν επεξεργαστή RISC όλες οι εντολές αναφοράς στη μνήμη (load/store) έχουν μέγεθος 32 bit, είναι ευθυγραμμισμένες (word aligned) στη μνήμη και αναφέρονται σε μία μόνο ευθυγραμμισμένη διεύθυνση μνήμης. Πόσες και ποιες σελίδες πρέπει να έχουμε στη μνήμη ώστε να μην μπορεί να συμβεί σφάλμα σελίδας κατά την εκτέλεση μίας τέτοιας εντολής; Τι αλλάζει αν οι εντολές μπορούν να ξεκινάνε σε οποιοδήποτε byte μιας λέξης των 32 bit (non word aligned);
8. [10] Εξηγήστε γιατί η συνεχόμενη εκχώρηση ή συνεχής κατανομή (continuous allocation) *τμημάτων/μπλοκ έχει καλύτερη απόδοση από τον πίνακα εκχώρησης ή κατανομής αρχείων (FAT) και τους κόμβους ευρετηρίου (i-nodes).* Εξηγήστε γιατί *δεν* χρησιμοποιείται στους *μαγνητικούς δίσκους και στους δίσκους SSD.*
9. *[10] Στα συστήματα αρχείων του UNIX,* για κάθε αρχείο διατηρούμε το χρόνο δημιουργίας, το χρόνο τελευταίας εγγραφής και το χρόνο τελευταίας προσπέλασης. Τι κόστος έχει η διατήρηση *κάθε ενός από αυτούς τους χρόνους,* δηλαδή, σε ποιες περιπτώσεις πρέπει να ενημερώνεται *(και να γράφεται στο δίσκο)* το i-node του αρχείου;
10. *[10] Γιατί η C είναι τόσο ευάλωτη στις* επιθέσεις υπερχείλισης στοίβας/ενταμιευτή/προσωρινής μνήμης (buffer overflow); Πώς μπορεί ένας προγραμματιστής να τις περιορίσει (χωρίς να αλλάξει γλώσσα προγραμματισμού);