ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Λειτουργικά Συστήματα

Εαρινό Εξάμηνο 2018-2019
Θέματα Εξετάσεων Ιουνίου – 12 Ιουνίου 2019
Εξεταστής: Γεώργιος Β. Ξυλωμένος

Οδηγίες:

- Διάρκεια εξέτασης: Δύο (2) ώρες.
- 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] Μπορεί ένα λειτουργικό σύστημα να κάνει χρονομερισμό (time sharing) χωρίς να έχει πολυπρογραμματισμό (multiprogramming); Μπορεί να κάνει πολυπρογραμματισμό χωρίς να έχει χρονομερισμό; Εξηγήστε τις απαντήσεις σας.
2. [10] Ποια κατηγορία διεργασιών οργανώνεται στις ουρές του χρονοπρογραμματιστή (scheduler), οι εμποδισμένες (blocked) ή οι έτοιμες (ready), και γιατί; Σε τι είδους ουρές οργανώνονται οι διεργασίες της άλλη κατηγορίας;
3. [10] Γιατί τα συστήματα χρονοπρογραμματισμού με σταθερές προτεραιότητες διεργασιών κινδυνεύουν από υποσιτισμό (starvation); Εξηγήστε γιατί το να δίνουμε μεγαλύτερα κβάντα στις χαμηλότερες προτεραιότητες δεν εγγυάται ότι δεν θα έχουμε υποσιτισμό.
4. [10] Σε ένα σύστημα με καταχωρητές βάσης και ορίου (base and index registers), ποιο πρόβλημα διαχείρισης μνήμης λύνουν οι καταχωρητές βάσης και ποιο πρόβλημα λύνουν οι καταχωρητές ορίου;
5. [10] Γιατί όταν χρησιμοποιούμε σκληρό δίσκο για αποθήκευση προσπαθούμε να έχουμε συνεχόμενη καταχώρηση των μπλοκ κάθε αρχείου, ακόμη κι όταν αυτό δεν είναι υποχρεωτικό από το σύστημα αρχείων; Γιατί στους δίσκους flash δεν μας απασχολεί αυτό;
6. [10] Στην κρυφή μνήμη μπλοκ (block cache) των συστημάτων αρχείων, ποια μπλοκ πρέπει να γράφονται πιο γρήγορα στο δίσκο όταν τροποποιούνται, τα μπλοκ των καταλόγων ή τα μπλοκ των απλών αρχείων, και γιατί;
7. [10] Στη διαχείριση αδιεξόδων, ο εντοπισμός και ανάκαμψη (detection and recovery) μπορεί να χρησιμοποιεί σχεδόν τον ίδιο αλγόριθμο με την αποφυγή αδιεξόδων (deadlock avoidance). Σε τι διαφέρουν λοιπόν οι δύο προσεγγίσεις;
8. [10] Στην ψηφιακή υπογραφή ενός αρχείου πρώτα χρησιμοποιούμε μία συνάρτηση κατακερματισμού στο αρχείο για να παράξουμε ένα μπλοκ σταθερού μήκους και μετά κρυπτογραφούμε αυτό το μπλοκ με το ιδιωτικό μας κλειδί. Γιατί δεν κρυπτογραφούμε όλο το αρχείο με το ιδιωτικό μας κλειδί;
9. [10] Γιατί οι επιθέσεις υπερχείλισης στοίβας (stack overflow) δεν μπορούν να εκτελούν κώδικα από τη στοίβα στα σύγχρονα συστήματα; Πώς παρακάμπτεται αυτό το πρόβλημα με τις επιθέσεις επιστροφής στη βιβλιοθήκη της C (return to libc);
10. [10] Στα συστήματα Linux ο χρήστης αρχικά δίνει το όνομα (username) και τον κωδικό (password) του σε μία διεργασία επιπέδου πυρήνα, αλλά αφού συνδεθεί στο σύστημα ο φλοιός (shell) του εκτελείται σε επίπεδο χρήστη. Γιατί η αλλαγή από επίπεδο σε πυρήνα σε επίπεδο χρήστη πρέπει να γίνει αφού ελεγχθεί ο κωδικός του χρήστη και όχι πριν;