Λειτουργικά Συστήματα
Ασκήσεις –Ερωτήσεις

Κεφάλαιο 1

Άσκηση 1

(α) Τι είναι ο πολυπρογραμματισμός (multiprogramming);

(β) Τι είναι η παροχέτευση εισόδου και εξόδου (input spooling / output spooling);

Απάντηση 1

(α) Τι είναι ο πολυπρογραμματισμός (multiprogramming); Σε τι βοηθάει;

- Πολυπρογραμματισμός (Multiprogramming) είναι η λειτουργία της ταχύτατης μετάβασης της CPU, μεταξύ πολλών διεργασιών που βρίσκονται στη μνήμη για εκτέλεση. Η χρήση του πολυπρογραμματισμού επιτρέπει στη CPU να έχει μεγαλύτερο βαθμό χρησιμοποίησης (utilization) και να μην βρίσκεται σε αδράνεια, κατά τη φάση που οι διεργασίες αναμένουν για Ε/Ε.

(β) Τι είναι η παροχέτευση εισόδου και εξόδου (input spooling / output spooling);

- Η παροχέτευση εισόδου είναι η λειτουργία της ανάγνωσης εργασιών (jobs) έτσι ώστε να επιτρέπεται η σειριακή εκτέλεσή τους και να μειωθεί ο χρόνος αναμονής της CPU. Η παροχέτευση εξόδου είναι η λειτουργία η οποία επιτρέπει τη δημιουργία μίας σειράς εργασιών εξόδου (π.χ. αποθήκευση αρχείων σε κάποια ουρά αναμονής πριν την εκτύπωσή τους).

Άσκηση 2

Ο βασικός λόγος που άργησαν να υιοθετηθούν τα GUI είναι το κόστος του υλικού που χρειαζόταν για την υποστήριξή τους.

(α) Πόση μνήμη RAM χρειάζεται για να υποστηρίξει μια μονόχρωμη οθόνη κειμένου με χωρητικότητα χαρακτήρων **25 γραμμές × 80 στήλες**;

(β) Πόση RAM χρειάζεται για μία **έγχρωμη οθόνη bitmap** με ανάλυση **1024 × 768 pixel των 24 bit**; Πόσο το κόστος σε μνήμη σε τιμές 1980 (5 $/KB μνήμης); Πόσο είναι το κόστος σήμερα (με μονάδα κόστους 10^{-5} $ / KB μνήμης);

Απάντηση 2

(α) 25×80 = 2000 χαρακτήρες. Αν κάθε χαρακτήρας είναι 1byte, τότε οι απαιτήσεις μνήμης RAM για την υποστήριξη της προβολής της οθόνης είναι περίπου 2KB μνήμης.

(β) Η οθόνη αποτελείται από 1024×768 = 786.432 pixel.

Κάθε pixel είναι 24 bit άρα 24/8 = 3 byte.

Συνεπώς 786.432×3 = 2.359.296 byte ή 2.304 KB.

Η απαιτούμενη μνήμη RAM σε τιμές 1980 είναι:

2.304 KB × 5 $/KB = 11.520$

Σε σημερινές τιμές: 2.304 KB × 10^{-5} $/KB = 0,02304$

Άσκηση 3

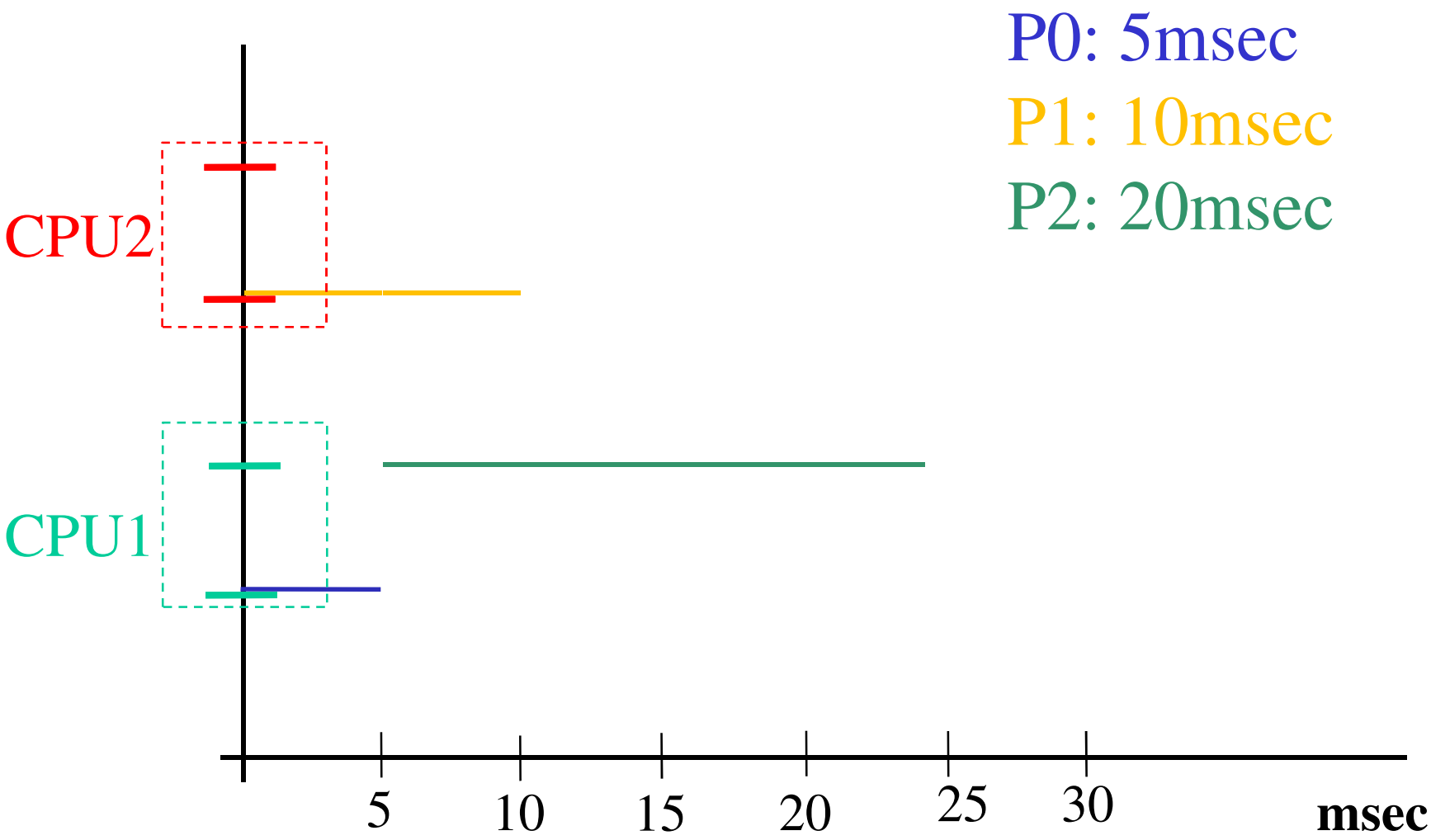
Θεωρήστε ένα σύστημα με **2 CPU**, στο οποίο κάθε CPU έχει **2** νήματα **(υπερνημάτωση – hyperthreading).** Υποθέστε ότι ξεκινούν τρία προγράμματα, τα *P0*, *P1*, και *P2*, με χρόνους εκτέλεσης **5, 10**, και **20 msec** αντίστοιχα.

Α) Πόσος χρόνος θα χρειαστεί για την ολοκλήρωση της εκτέλεσης και των τριών προγραμμάτων;

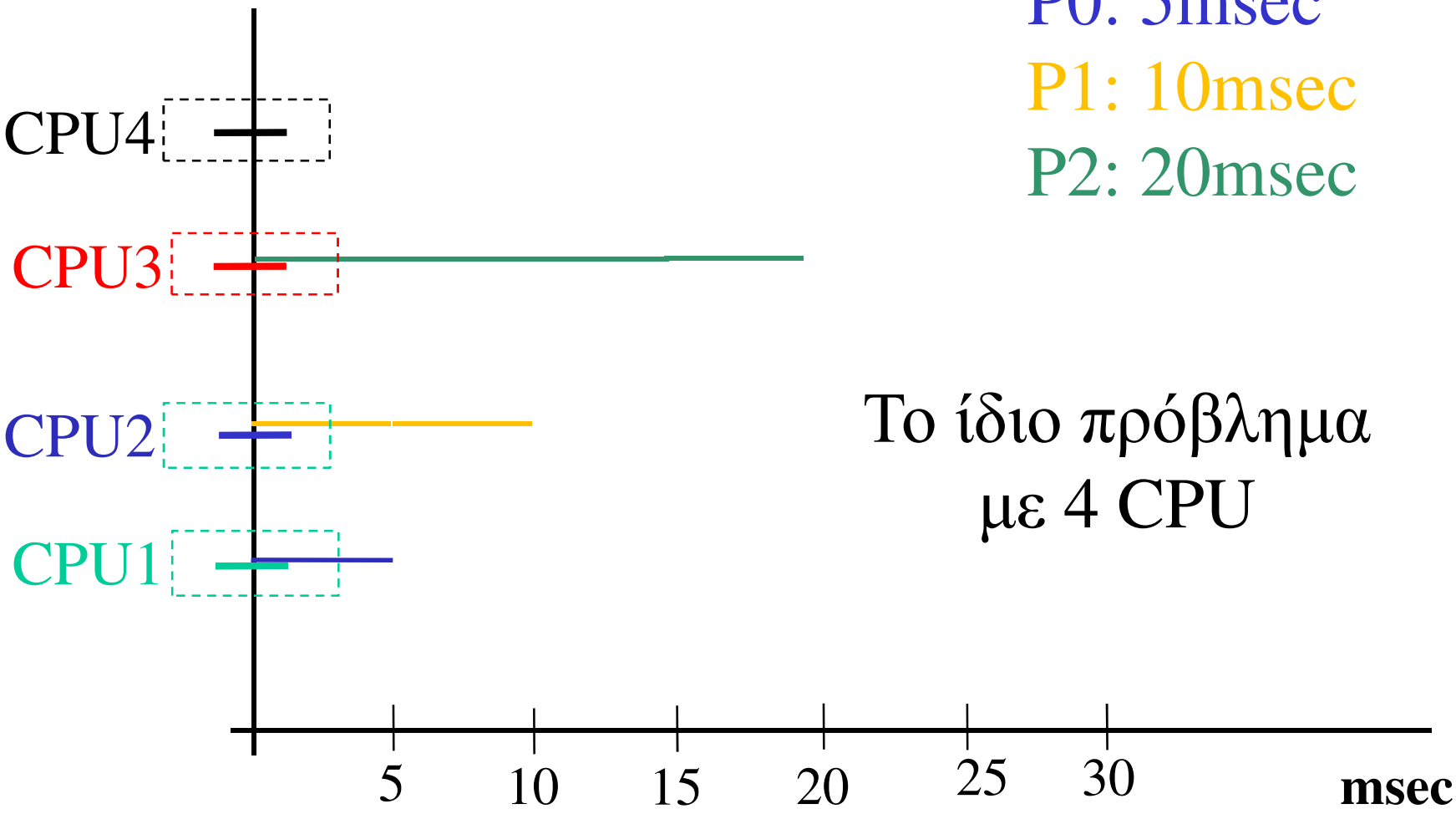
Β) Πόσος χρόνος θα χρειαζόταν με **4 CPU**;

Υποθέστε ότι και τα 3 προγράμματα χρειάζονται κατά 100% τη CPU, δεν μπλοκάρονται κατά την εκτέλεση, και δεν αλλάζουν CPU από τη στιγμή που θα τους ανατεθεί κάποια.

Απάντηση 3



Απάντηση 3 (...συνέχεια)



Το ίδιο πρόβλημα
με 4 CPU

Άσκηση 4

Θεωρήστε ένα σύστημα υπολογιστή που διαθέτει κρυφή μνήμη (cache memory), κύρια μνήμη (RAM) και δίσκο, και του οποίου το ΛΣ χρησιμοποιεί εικονική μνήμη (virtual memory). Χρειάζονται:

- **2 nsec** για την προσπέλαση μιας λέξης από την **cache memory,**
- **10 nsec** για την προσπέλαση μιας λέξης από τη **μνήμη RAM, και**
- **10 ms** για την προσπέλαση μιας λέξης από το **δίσκο.**

Αν το ποσοστό ευστοχίας της κρυφής μνήμης είναι **95%** και το ποσοστό ευστοχίας της κύριας μνήμης (μετά από μια αστοχία κρυφής μνήμης) είναι **99%**, ποιος είναι ο μέσος χρόνος προσπέλασης μιας λέξης;

Απάντηση 4

1 msec = 10^{-3} sec, 1 μsec = 10^{-6} sec, 1 nsec = 10^{-9} sec

Εφόσον στο 95% πετυχαίνει η cache memory, υπολείπεται 5% για τις άλλες δύο περιπτώσεις.

Η RAM έχει ευστοχία 99%, άρα στο υπολειπόμενο 1% οδηγούμαστε στο δίσκο.

2 nsec×95% + 5% [(10 nsec×99%) + (10 ms×1%)] =

1,9 nsec + 0,495 nsec + 0,005 ms =

2,395 nsec + 5 **μ**sec = 2,395 nsec + 5×10^3 **n**sec =

= 5.002,395nsec

Άσκηση 5

(α) Τι είναι μια **εντολή παγίδευσης (trap)**; Εξηγήστε τη χρήση της στα λειτουργικά συστήματα.

(β) Ποιά είναι η κυριότερη διαφορά ανάμεσα στις παγιδεύσεις και τις διακοπές (interrupts);

Απάντηση 5

(α) Μία παγίδευση προκαλείται από μία ***κλήση συστήματος*** (system call) ώστε να επιτρέψει σε ένα πρόγραμμα **να αλλάξει την κατάσταση της εκτέλεσής του** από την **κατάσταση χρήστη** (user mode) σε **κατάσταση πυρήνα** (kernel mode). Με αυτό τον τρόπο μπορεί να εκτελέσει λειτουργίες που απαιτούν ειδικά δικαιώματα (π.χ. Ε/Ε)

(β) Η παγίδευση (trap) προκαλείται από το ίδιο το πρόγραμμα, σε συγκεκριμένο(α) σημείο(α) του κώδικα. Συνεπώς, όσες φορές και εάν εκτελεστεί το πρόγραμμα, η παγίδευση θα συμβαίνει ακριβώς στην ίδια εντολή. Μία διακοπή (interrupt) προκαλείται από ένα εξωτερικό γεγονός (π.χ. ο χρονοπρογραμματιστής του ΛΣ) και οι χρονικές στιγμές που συμβαίνει δεν είναι προβλέψιμες.

Άσκηση 6

Ποιός είναι ο σκοπός των κλήσεων συστήματος σε ένα λειτουργικό σύστημα;

Απάντηση 6

Το ΛΣ είναι το μόνο λογισμικό το οποίο εκτελείται σε κατάσταση πυρήνα (kernel mode). Όλο το υπόλοιπο λογισμικό εκτελείται σε κατάσταση χρήστη (user mode).

Για να χρησιμοποιήσει μία διεργασία κάποια υπηρεσία του ΛΣ (π.χ. ανάγνωση από αρχείο) **εκτελεί την αντίστοιχη κλήση συστήματος (system call).**

Η κλήση συστήματος προκαλεί μία παγίδευση (trap) και μεταβαίνει σε κατάσταση πυρήνα. Η εντολή TRAP προκαλεί την ανάμιξη του ΛΣ.

Κάθε κλήση συστήματος υλοποιείται μέσω μίας ρουτίνας που βρίσκεται σε μία βιβλιοθήκη διαδικασιών (συνήθως σε C).

Άσκηση 7

Για κάθε μία από τις επόμενες κλήσεις συστήματος, δώστε μια συνθήκη που τις αναγκάζει να αποτύχουν:

- fork,
- exec, και
- unlink.

Απάντηση 7

Η κλήση συστήματος **fork** μπορεί να αποτύχει αν δεν υπάρχουν ελεύθερες θέσεις στον πίνακα διεργασιών (process table) (άρα δεν υπάρχει διαθέσιμος χώρος για μία νέα διεργασία).

Η κλήση συστήματος **exec** μπορεί να αποτύχει αν το όνομα του αρχείου που έχει δοθεί δεν υπάρχει ή εάν δεν αντιστοιχεί σε ένα έγκυρο εκτελέσιμο αρχείο.

Η κλήση συστήματος **unlink** μπορεί να αποτύχει αν το αρχείο προς αποσύνδεση δεν υπάρχει ή εάν η καλούσα διεργασία δεν έχει εξουσιοδότηση για να το αποσυνδέσει (π.χ. ανήκει σε άλλο χρήστη).

Άσκηση 8

Τι επιστρέφει η παρακάτω κλήση συστήματος;

counter = write(fd, &buffer, nbytes);

Μπορεί να επιστρέψει διαφορετική τιμή; Αν ναι, για ποιο λόγο;

Απάντηση 8

Επιστρέφει τον αριθμό των byte που πραγματικά αντιγράφτηκαν. Υπό κανονικές συνθήκες, επιστρέφει την τιμή nbytes.

Εάν η κλήση αποτύχει, για παράδειγμα επειδή ο περιγραφέας του αρχείου **fd** είναι λανθασμένος, τότε θα επιστρέψει την τιμή -1.

Μπορεί επίσης να αποτύχει γιατί ο δίσκος είναι γεμάτος και δεν είναι δυνατό να εγγράψει τον αριθμό των bytes που έχουν ζητηθεί.

Άσκηση 9

Ένα αρχείο μεγέθους **10 MB** είναι αποθηκευμένο σε ένα δίσκο στην ίδια τροχιά (τροχιά 50) σε διαδοχικούς τομείς. Ο βραχίονας του δίσκου βρίσκεται προς το παρόν, πάνω από την τροχιά 100. Πόσος χρόνος χρειάζεται για την ανάγνωση όλου του αρχείου από το δίσκο;

Υποθέσεις:

(1) Μετακίνηση του βραχίονα από ένα κύλινδρο στον επόμενο: **1 ms.**

(2) Περιστροφή του τομέα που περιέχει την αρχή του αρχείου κάτω από την κεφαλή: **5 ms.**

(3) Ταχύτητα ανάγνωσης: **100 MB/sec.**

Απάντηση 9

- Μετακίνηση από track 100 → track 50:
  50 × 1 ms = 50 ms.

- Περιστροφή του track 100 στην αρχή του αρχείου: **5 ms.**

- Χρόνος ανάγνωσης 10 MB με ταχύτητα 100 MB/sec:
  10 MB / (100 MB/sec) = 0,1 sec = 0,1 × 10^3 ms =100ms

Συνεπώς συνολικός χρόνος: **(50+5+100) ms = 155 ms**

Κεφάλαιο 2

Άσκηση 10

(α) Τι κάνει η κλήση συστήματος ***fork*()**;

(β) Τι είναι μία πολυνηματική διεργασία; Δώστε ένα παράδειγμα.

(γ) Όταν μια πολυνηματική διεργασία ενεργοποιεί την κλήση ***fork*( )**, ενδέχεται να παρουσιαστεί το εξής πρόβλημα.

Υποθέστε ότι ένα από τα αρχικά νήματα περίμενε για είσοδο από το πληκτρολόγιο. Η θυγατρική διεργασία λαμβάνει αντίγραφα όλων των νημάτων της γονικής. Τώρα περιμένουν *δύο νήματα για είσοδο από το πληκτρολόγιο*, ένα σε κάθε διεργασία.

Παρουσιάζεται το πρόβλημα αυτό σε μονονηματικές διεργασίες;

Απάντηση 10

(α) Η κλήση fork() χρησιμοποιείται όταν μία διεργασία θέλει να δημιουργήσει μία νέα διεργασία. Με την κλήση της fork() δημιουργείται ένα ακριβές αντίγραφο της καλούσας διεργασίας.

(β) Μία πολυνηματική διεργασία (multithreaded process) είναι μία διεργασία η οποία περιλαμβάνει πολλές διαφορετικές ροές ελέγχου. Π.χ., ένας σύγχρονος επεξεργαστής κειμένου ενδέχεται να έχει ένα νήμα για την λήψη εισόδου από το πληκτρολόγιο, ένα νήμα για την διαρκή σελιδοποίηση, ένα άλλο για αυτόματη αποθήκευση κ.τ.λ.

(γ) Δεν δημιουργείται ποτέ αυτό το πρόβλημα στις μονονηματικές διεργασίες για τον εξής λόγο. Αν μία μονονηματική διεργασία περιμένει Ε/Ε από το πληκτρολόγιο (άρα έχει ***μπλοκαριστεί***), ***δεν μπορεί να καλέσει την fork***.

Άσκηση 11

(α)Για τον χρονοπρογραμματισμό νημάτων χρησιμοποιείται η κλήση *thread_yield*(); Τι ακριβώς κάνει;

(β)Γιατί θα ήθελε ένα νήμα να επιστρέψει εκούσια τον έλεγχο της CPU καλώντας τη *thread_yield*; Σε τελική ανάλυση, από τη στιγμή που δεν υπάρχουν περιοδικές διακοπές ρολογιού, υπάρχει πιθανότητα να μην ξαναπάρει ποτέ τον έλεγχο της CPU.

Απάντηση 11

(α) Η κλήση thread_yield καλείται από ένα νήμα που εκτελείται, ώστε να **παραχωρήσει εθελοντικά** τη σειρά του σε κάποιο άλλο νήμα της ίδιας διεργασίας.

(β) Τα νήματα μίας διεργασίας θα πρέπει να συνεργάζονται, εφόσον **λειτουργούν στο πλαίσιο της ίδιας διεργασίας** και του ίδιου προγράμματος το οποίο συνήθως έχει γραφτεί από ένα προγραμματιστή (ή μία ομάδα προγραμματιστών) **για ένα συγκεκριμένο λόγο.**

Συνεπώς, εάν ο προγραμματιστής κρίνει ότι ένα νήμα θα πρέπει να έχει προτεραιότητα, τα άλλα νήματα θα είναι προγραμματισμένα να καλούν την yield.

Άσκηση 12

Είναι δυνατόν να προεκτοπιστεί ένα νήμα από διακοπή ρολογιού; Αν ναι, κάτω από ποιές προϋποθέσεις; Αν όχι, γιατί;

Απάντηση 12

Τα νήματα ***επιπέδου χρήστη δεν προεκτοπίζονται από τις διακοπές ρολογιού***. Ο χρονοπρογραμματισμός τους γίνεται σε *επίπεδο προγράμματος χρήστη*, ενώ για ολόκληρη τη διεργασία ο χρονοπρογραμματισμός γίνεται σε επίπεδο πυρήνα. Η διεργασία λαμβάνει ένα κβάντο χρόνου στη CPU και αποφασίζει μόνη της πώς θα το μοιράσει στα νήματά της.

Η διακοπή των νημάτων από διακοπή ρολογιού ***είναι δυνατή μόνο για τα νήματα επιπέδου πυρήνα.*** Σε αυτή την περίπτωση, κάθε νήμα χρονοπρογραμματίζεται όπως μία διεργασία με ένα μόνο νήμα.

Άσκηση 13

Να συγκρίνετε την ανάγνωση ενός αρχείου με τη χρήση **μονονηματικού διακομιστή** αρχείων (single-threaded server) και με τη χρήση **πολυνηματικού διακομιστή** (multi-threaded-server).

- Εάν τα δεδομένα που απαιτούνται βρίσκονται στην κρυφή μνήμη (cache hit), χρειάζονται 15 msec για να ληφθεί μια αίτηση εργασίας, να διεκπεραιωθεί, και να γίνει η υπόλοιπη επεξεργασία.
- Στο 1/3 των περιπτώσεων χρειάζεται λειτουργία δίσκου, και απαιτούνται επιπλέον 75 msec, στη διάρκεια των οποίων το νήμα είναι σε λήθαργο.

Ποια η διεκπεραιωτική ικανότητα (αιτήσεις ανά sec) για έναν μονονηματικό και ποια για έναν πολυνηματικό διακομιστή;

Απάντηση 13

- Μονονηματικός εξυπηρετητής:
  - Όταν συμβαίνει ευστοχία κρυφής μνήμης, η διεκπεραίωση μίας αίτησης απαιτεί 15 msec.
  - Η αστοχία κρυφής μνήμης και η εξυπηρέτηση μέσω του δίσκου, απαιτεί 75 msec.
  - Ο μέσος χρόνος εξυπηρέτησης είναι:

    (2/3×15 msec) + (1/3×90 msec) = 40 msec, άρα **25 αιτήσεις /sec**.
- Πολυνηματικός εξυπηρετητής:
  - Σε αυτή την περίπτωση, όταν ένα νήμα αναμένει Ε/Ε από το δίσκο, κάποιο άλλο νήμα μπορεί να εξυπηρετεί ένα άλλο αίτημα.
  - Συνεπώς η αναμονή για Ε/Ε από το δίσκο επικαλύπτεται.
  - Κάθε αίτηση διεκπαιρεώνεται σε 15 msec, και ο server εξυπηρετεί **66 2/3 αιτήσεις/sec.**

Άσκηση 14

Μπορεί να μετρηθεί αν μια διεργασία είναι ***εξαρτημένη από τη CPU*** ή ***εξαρτημένη από Ε/Ε***, αν αναλυθεί ο πηγαίος κώδικάς της;

Μπορεί αυτό να προσδιοριστεί κατά το χρόνο εκτέλεσης του προγράμματος;

Απάντηση 14

Σε απλές περιπτώσεις μπορούμε να δούμε εάν ένα πρόγραμμα είναι εξαρτημένο από Ε/Ε. Π.χ. ***ένα πρόγραμμα το οποίο διαβάζει στην αρχή όλα τα αρχεία εισόδου*** και τοποθετεί τα δεδομένα που χρειάζεται σε buffers ***πιθανότατα δεν θα είναι εξαρτημένο από Ε/Ε.***

Ένα πρόγραμμα που διαβάζει και διαβάζει συνέχεια διαφορετικά αρχεία (π.χ. ένας μεταγλωττιστής) ***πιθανότατα θα είναι εξαρτημένο από Ε/Ε.***

Κατά τη διάρκεια του χρόνου εκτέλεσης μπορούμε (κατά προσέγγιση) να δούμε ***εάν το πρόγραμμα είναι εξαρτημένο από τη CPU*** με τη ***χρήση π.χ. της εντολής ps*** *(βλέποντας τι ποσοστό της CPU έχει χρησιμοποιήσει ένα πρόγραμμα, σε σύγκριση με το συνολικό χρόνο εκτέλεσης του προγράμματος).*

Άσκηση 15

Μετρήσεις σε συγκεκριμένο σύστημα έχουν δείξει ότι ο μέσος όρος του χρόνου εκτέλεσης (CPU time) για μια διεργασία είναι ***T***, πριν αυτή μπλοκαριστεί από είσοδο/έξοδο. Η εναλλαγή διεργασίας (context switch time) απαιτεί χρόνο ***S***, ο οποίος λογίζεται ως επιβάρυνση (χαμένος χρόνος). Εάν έχουμε χρονοπρογραμματισμό εκ περιτροπής (*Round-Robin*) με κβάντο χρόνου ***Q***, δώστε το μαθηματικό τύπο για την ***αποδοτικότητα (efficiency) της CPU*** σε κάθε μία από τις εξής περιπτώσεις.

(α) $Q = \infty$

(β) $Q > T$

(γ) $S < Q < T$

(δ) $Q = S$

(ε) $Q \approx 0$

Απάντηση 15

$$\text{Αποδοτικότητα CPU} = \frac{\text{Χρήσιμος χρόνος CPU}}{\text{Συνολικός χρόνος CPU}}$$

1. Εάν $Q \geq T$, τότε κάθε διεργασία θα τρέξει για **T** και μετά θα γίνει εναλλαγή διεργασίας για χρόνο **S**. Άρα η αποδοτικότητα είναι: $\text{Αποδοτικότητα} = \frac{T}{T+S}$

1. Εάν $Q<T$, κάθε διεργασία θα χρειαστεί **T/Q** εναλλαγές διεργασιών, σπαταλώντας χρόνο **S×(T/Q)**. Άρα η αποδοτικότητα είναι: $\text{Αποδοτικότητα} = \frac{T}{T+S\cdot(T/Q)}$

...Απάντηση 15

Με βάση την περίπτωση (1), ισχύει για τα (α), (β):

(α) $Q = \infty$

(β) $Q > T$

$$Αποδ. = \frac{T}{T+S}$$

Με βάση την περίπτωση (2) ισχύει για τα (γ), (δ), (ε):

(γ) $S < Q < T$ $$\text{Αποδ.} = \frac{T}{T+S\cdot(T/Q)} = \frac{T}{T(1+S/Q)} = \frac{1}{\frac{Q+S}{Q}} = \frac{Q}{Q+S}$$

(δ) $Q = S\ (<T)$ $$\text{Αποδ.} = \frac{T}{T+S\cdot(T/Q)} = \frac{T}{T+T} = \frac{1}{2}$$

(ε) $Q \approx 0$. Εφόσον το Q→0, τότε T/Q→∞ και **Αποδ.**→ 0.

Άσκηση 16

Πέντε εργασίες περιμένουν να εκτελεστούν. Οι αναμενόμενοι χρόνοι εκτέλεσης τους είναι 9, 6, 3, 5, και Χ.

Με *ποιο αλγόριθμο* και με *ποιά σειρά* πρέπει να εκτελεστούν ώστε να ελαχιστοποιηθεί ο μέσος χρόνος απόκρισης;

(Η απάντηση πρέπει να είναι συνάρτηση του Χ.)

Απάντηση 16

Ο ***αλγόριθμος SJF*** (shortest job first) δίνει το ***μικρότερο μέσο χρόνο απόκρισης***. Συνεπώς θα ισχύει:

- Αν $0 < X \leq 3$: **X**, 3, 5, 6, 9.
- Αν $3 < X \leq 5$: 3, **X**, 5, 6, 9.
- Αν $5 < X \leq 6$: 3, 5, **X**, 6, 9.
- Αν $6 < X \leq 9$: 3, 5, 6, **X**, 9.
- Αν $X > 9$: 3, 5, 6, 9, **X**.

Άσκηση 17

Εργασίες δέσμης Α, Β, Γ, Δ, Ε, καταφθάνουν σε ένα κέντρο υπολογιστών την ίδια περίπου χρονική στιγμή. Οι χρόνοι εκτέλεσής τους εκτιμώνται σε 10, 6, 2, 4, και 8 λεπτά αντίστοιχα.

Οι προτεραιότητές τους (που καθορίστηκαν εξωτερικά) είναι 3, 5, 2, 1, και 4 αντίστοιχα, (όπου 5 η υψηλότερη).

Για τον καθέναν από τους επόμενους αλγόριθμους χρονοπρογραμματισμού, υπολογίστε το μέσο χρόνο διεκπεραίωσης των διεργασιών.

Αγνοήστε την επιβάρυνση λόγω εναλλαγής των διεργασιών.

...

...Άσκηση 17

... (α) Εξυπηρέτηση εκ περιτροπής.

(β) Χρονοπρογραμματισμός προτεραιοτήτων.

(γ) Εξυπηρέτηση με βάση τη σειρά άφιξης (εκτέλεση με τη σειρά 10, 6, 2, 4, 8).

(δ) Εξυπηρέτηση με βάση τη μικρότερη διάρκεια.

Για την περίπτωση (α), υποθέστε ότι το σύστημα είναι πολυπρογραμματιζόμενο και ότι κάθε εργασία παίρνει **δίκαιο μερίδιο του χρόνου** της CPU.

Για τις περιπτώσεις (β) έως (δ), υποθέστε ότι ο αλγόριθμος είναι **μη προεκτοπιστικός**.

Όλες οι εργασίες είναι τελείως εξαρτημένες από τη CPU.

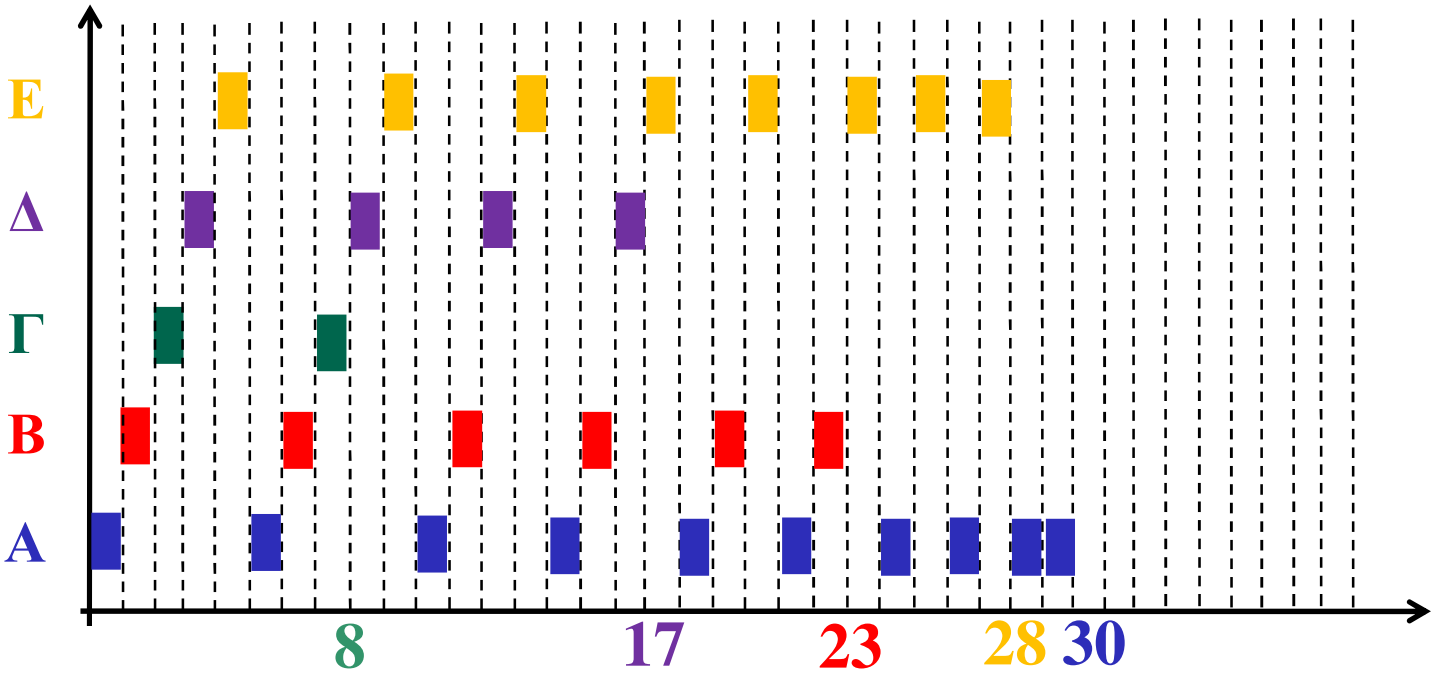
Απάντηση 17 (α)

Α: 10, Β: 6, Γ: 2, Δ: 4, και Ε: 8.

(α) RR με δίκαιη κατανομή χρόνου.

Ο μέσος χρόνος εκτέλεσης είναι:

(8+17+23+28+30)/5 = 106/5 = 21,2 λεπτά.



...Απάντηση 17(β)

Α: 10, Β: 6, Γ: 2, Δ: 4, και Ε: 8.

Προτεραιότητες **3, 5, 2, 1, και 4.**

(β) Χρονοπρογραμματισμός προτεραιοτήτων μη-προεκτοπιστικός.

Η **Β**: τερματίζει σε 6 λεπτά.

Η **Ε**: τερματίζει σε 6+8 = 14 λεπτά.

Η **Α**: τερματίζει σε 14+10 = 24 λεπτά.

Η **Γ**: τερματίζει σε 24+2 = 26 λεπτά.

Η **Δ**: τερματίζει σε 26+4 = 30 λεπτά.

Ο μέσος χρόνος εκτέλεσης είναι:

(6+14+24+26+30)/5 = 100/5 = 20 λεπτά.

...Απάντηση 17(γ)

Α: 10, Β: 6, Γ: 2, Δ: 4, και Ε: 8.

(γ) Εξυπηρέτηση με βάση τη σειρά άφιξης .

Η **Α**: τερματίζει σε 10 λεπτά.

Η **Β**: τερματίζει σε 10+6 = 16 λεπτά.

Η **Γ**: τερματίζει σε 16+2 = 18 λεπτά.

Η **Δ**: τερματίζει σε 18+4 = 22 λεπτά.

Η **Ε**: τερματίζει σε 22+8 = 30 λεπτά.

Ο μέσος χρόνος εκτέλεσης είναι:

(10+16+18+22+30)/5 = 96/5 = 19,2 λεπτά.

...Απάντηση 17 (δ)

Α: 10, Β: 6, Γ: 2, Δ: 4, και Ε: 8.

(δ) Εξυπηρέτηση με βάση τη μικρότερη διάρκεια (SJF).

Η **Γ**: τερματίζει σε 2 λεπτά.

Η **Δ**: τερματίζει σε 2+4 = 6 λεπτά.

Η **Β**: τερματίζει σε 6+6 = 12 λεπτά.

Η **Ε**: τερματίζει σε 12+8 = 20 λεπτά.

Η **Α**: τερματίζει σε 20+10 = 30 λεπτά.

Ο μέσος χρόνος εκτέλεσης είναι:

(2+6+12+20+30)/5 = 70/5 = 14 λεπτά.

Κεφάλαιο 3

Άσκηση 18

Θεωρήστε ένα σύστημα εναλλαγής (memory swap), στο οποίο η μνήμη περιέχει τα παρακάτω μεγέθη κενών (οπών) κατά σειρά:

10 KB, 4 KB, 20 KB, 18 KB, 7 KB, 9 KB, 12 KB, και 15 KB.

Ποιά οπή θα χρησιμοποιηθεί αν γίνουν συνεχόμενες αιτήσεις τμημάτων με μέγεθος

(α) 12 KB

(β) 10 KB

(γ) 9 KB

και χρησιμοποιείται ***ο αλγόριθμος της πρώτης προσαρμογής***; Επαναλάβετε την άσκηση για τους αλγόριθμους της ***βέλτιστης προσαρμογής***, και της **χείριστης προσαρμογής.**

Απάντηση 18

- Πρώτη προσαρμογή: 20 KB, 10 KB, 18 KB.
- Βέλτιστη προσαρμογή: 12 KB, 10 KB, 9 KB.
- Χείριστη προσαρμογή: 20 KB, 18 KB, 15 KB.

Άσκηση 19

Ένας υπολογιστής με διευθύνσεις των **32 bit** χρησιμοποιεί έναν πίνακα σελίδων δύο επιπέδων. Οι εικονικές διευθύνσεις χωρίζονται σε ένα πεδίο **9 bit** για τον πίνακα σελίδων του υψηλότερου επιπέδου, ένα πεδίο **11 bit** για τον πίνακα σελίδων του δεύτερου επιπέδου, και μια σχετική διεύθυνση (offset).

Πόσο μεγάλες είναι οι σελίδες και πόσες υπάρχουν στο χώρο διευθύνσεων;

Απάντηση 19

- Συνολικά, 20 bit χρησιμοποιούνται για τις εικονικές διευθύνσεις, αφήνοντας 12 για την σχετική διεύθυνση (offset).

- Αυτό σημαίνει ότι κάθε σελίδα έχει μέγεθος 2^{12} bit ή $2^{12}/2^3$ byte ή $2^{12}/(2^3\ 2^{10}) = ½$ KB.

- Ένας χώρος εικονικών διευθύνσεων των 20 bit έχει 2^{20} σελίδες.

Άσκηση 20

Ένας υπολογιστής με διεργασίες που έχουν **1024 σελίδες** στο χώρο διευθύνσεών τους, διατηρεί τους πίνακες σελίδων του στη μνήμη.

Η επιβάρυνση που απαιτείται για να διαβαστεί μια λέξη από τον πίνακα σελίδων είναι **5 nsec.** Για να μειωθεί αυτή η επιβάρυνση, ο υπολογιστής διαθέτει μία **TLB** η οποία διατηρεί **32 καταχωρίσεις** (εικονική σελίδα, φυσικό πλαίσιο σελίδας) και μπορεί να κάνει μία αναζήτηση σε **1 nsec.**

Τι ποσοστό ευστοχίας απαιτείται ώστε να μειωθεί η μέση επιβάρυνση σε **2 nsec;**

Απάντηση 20

Θα πρέπει 1*α + 5* (1-α) = 2 ➔ α = 3/4.

Άσκηση 21

Μια μηχανή έχει **εικονικές διευθύνσεις 48 bit** και **φυσικές διευθύνσεις 32 bit**.

Οι σελίδες έχουν μέγεθος **8 KB.**

Πόσες εικονικές σελίδες υπάρχουν;

Πόσες καταχωρίσεις απαιτούνται για τον πίνακα σελίδων;

Πόσα πλαίσια σελίδων υπάρχουν;

Απάντηση 21

- Μέγεθος σελίδας 8KB = $2^3 * 2^{10} * 2^3 = 2^{16}$ bit.
- Χώρος εικονικών διευθύνσεων 48 bit, άρα 2^{48} εικονικές διευθύνσεις.
- Χώρος φυσικών διευθύνσεων 32 bit, άρα 2^{32} φυσικές διευθύνσεις.

1) Υπάρχουν συνολικά $2^{48} / 2^{16} = 2^{32}$ σελίδες (περίπου 34 δισ.)
2) Οι καταχωρήσεις του πίνακα σελίδων είναι τόσες, όσες είναι οι οι εικονικές σελίδες (2^{32})
3) Τα πλαίσια σελίδας είναι συνολικά $2^{32} / 2^{16} = 2^{16}$ πλαίσια (65.536)

Άσκηση 22

Αν ο αλγόριθμος αντικατάστασης σελίδας FIFO χρησιμοποιηθεί με τέσσερα πλαίσια σελίδας και οκτώ σελίδες, πόσα σφάλματα σελίδας θα προκύψουν αν η συμβολοσειρά αναφορών είναι 0-1-7-2-3-2-7-1-0-3, με δεδομένο ότι τα τέσσερα πλαίσια είναι αρχικά άδεια;

Επαναλάβετε το ίδιο πρόβλημα για τον αλγόριθμο LRU (υπόδειξη: θεωρείστε ότι κάθε καταχώρηση του πίνακα σελίδων, διατηρεί ένα μετρητή Μ όπου δείχνει το πλήθος των εντολών κατά τη χρονική στιγμή της τελευταίας αναφοράς στη σελίδα)

Απάντηση 22

Για τον αλγόριθμο FIFO προκύπτουν 6 σφάλματα:

0-1-7-2-3-2-7-1-0-3

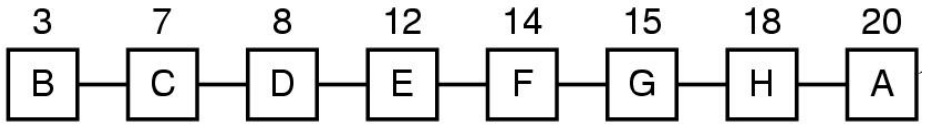
x0**x**1**x**7**x**2**x3271x03**

Για τον αλγόριθμο LRU προκύπτουν 7 σφάλματα:
x0**x**1**x**7**x**2**x**3271**x**0**x**3

Άσκηση 23

Θεωρήστε την ακολουθία σελίδων της παρακάτω εικόνας. Υποθέστε ότι τα bit **Αναφοράς (A)** για τις σελίδες *B* έως *A* είναι 11011011, αντίστοιχα.

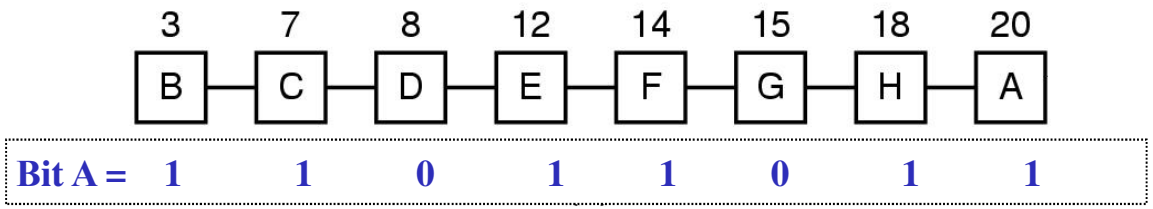
Ποια σελίδα θα αφαιρέσει ο αλγόριθμος της δεύτερης ευκαιρίας;



(b)

Απάντηση 23

Θα επιλεγεί η 1η σελίδα με bit A=0, άρα η D



Άσκηση 24

Ένας μικρός υπολογιστής έχει τέσσερα πλαίσια σελίδας. Στον πρώτο χτύπο του ρολογιού τα bit *A* έχουν τις τιμές 0111 (η σελίδα 0 έχει την τιμή 0, ενώ οι υπόλοιπες 1). Στους επόμενους χτύπους ρολογιού, οι τιμές είναι 1011, 1010, 1101, 0010, 1010, 1100, και 0001.

Αν χρησιμοποιείται ο **αλγόριθμος γήρανσης** με ένα μετρητή 8 bit, βρείτε τις τιμές των τεσσάρων μετρητών μετά από τον τελευταίο χτύπο ρολογιού.

Απάντηση 24

Εφαρμόζουμε τον αλγόριθμο γήρανσης στις σελίδες (πρόσθεση του bit Α και μετά ολίσθηση δεξιά. Οι μετρητές θα είναι τελικά:

Σελίδα 0: 01101100

Σελίδα 1: 01001001

Σελίδα 2: 00110111

Σελίδα 3: 10001011

Άσκηση 25

Ένας υπολογιστής έχει τέσσερα πλαίσια σελίδας. Ο χρόνος φόρτωσης, ο χρόνος τελευταίας προσπέλασης, και οι τιμές των bit *A* και *T* για κάθε σελίδα φαίνονται παρακάτω (οι χρονικές στιγμές είναι σε χτύπους ρολογιού):

Σελίδα	Φορτώθηκε	Τελευταία αναφορά	A	T
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

(α) Ποια σελίδα θα αντικαταστήσει ο αλγόριθμος NRU;
(β) Ποια σελίδα θα αντικαταστήσει ο αλγόριθμος FIFO;
(γ) Ποια σελίδα θα αντικαταστήσει ο αλγόριθμος LRU;
(δ) Ποια σελίδα θα αντικαταστήσει ο αλγόριθμος της δεύτερης ευκαιρίας;

Απάντηση 25

Ο NRU τη σελίδα 2 (λόγω των bit A, T)

Ο FIFO τη σελίδα 3 (λόγω παλαιότερου χρόνου φόρτωσης)

Ο LRU τη σελίδα 1 (αναφέρθηκε λιγότερο πρόσφατα)

Ο αλγόριθμος της δεύτερης ευκαιρίας τη σελίδα 2 (η 3 είναι παλαιότερη αλλά το bit A της 3 είναι 1 ενώ το bit A της 2 είναι 0)

Σελίδα	Φορτώθηκε	Τελευταία αναφορά	A	T
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

Κεφάλαιο 4

Άσκηση 26

Ένας τρόπος για να χρησιμοποιηθεί συνεχής κατανομή στο δίσκο χωρίς το πρόβλημα που προκαλείται από τα κενά είναι να συμπτύσσεται ο δίσκος κάθε φορά που αφαιρείται ένα αρχείο. Από τη στιγμή που όλα τα αρχεία είναι συνεχή, η αντιγραφή ενός αρχείου απαιτεί μία αναζήτηση και μία καθυστέρηση λόγω περιστροφής κατά την ανάγνωση του αρχείου, οι οποίες ακολουθούνται από τη μεταφορά (που γίνεται με τη μέγιστη ταχύτητα). Η εγγραφή του αρχείου απαιτεί τις ίδιες εργασίες.

Αν υποθέσουμε ότι ο χρόνος αναζήτησης είναι **5 msec**, η καθυστέρηση λόγω περιστροφής **4 msec**, ο ρυθμός μεταφοράς **8 MB/sec**, και το μέσο μέγεθος αρχείου **8 KB**, πόσος χρόνος χρειάζεται για να διαβαστεί ένα αρχείο στην κύρια μνήμη και στη συνέχεια να γραφεί ξανά στο δίσκο σε νέα θέση;

Με βάση τους ίδιους αριθμούς, πόσος χρόνος χρειάζεται για να συμπτυχθεί ο μισός χώρος ενός δίσκου **16 GB**;

Απάντηση 26

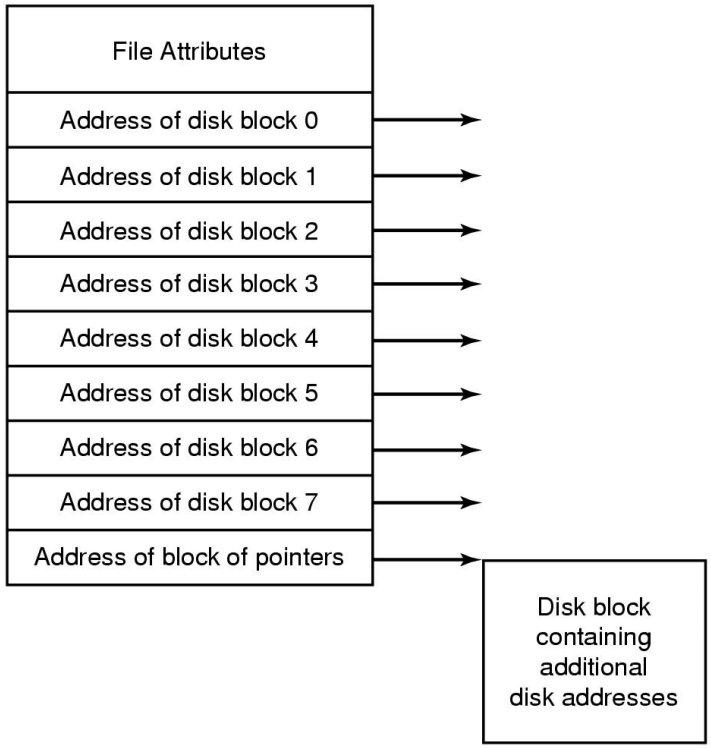
- Για να ξεκινήσει η ανάγνωση, χρειάζονται: 5+4 =9 msec.
- Για την ανάγνωση 8 KB=2^{13} bytes με ταχύτητα μεταφοράς 8 MB/sec = 2^{23} bytes/sec χρειάζονται 2^{-10} sec (περ. 0,9765 msec).
- Συνολικός χρόνος ανάγνωσης 9,9765 msec..
- Συνολικός χρόνος ανάγνωσης και επανεγγραφής:

  2 * 9,9765 = 19.953 msec.

- Η συμπίεση του μισού δίσκου συνολικού μεγέθους 16 GB, απαιτεί την εγγραφή 8 GB, (δηλ. με μέγεθος αρχείων 8 KB, σύνολο 2^{33} / 2^{13} = 2^{20} αρχεία).
- Χρόνος συμπίεσης 2^{20} * 19.953 msec περίπου 5,8 ώρες.

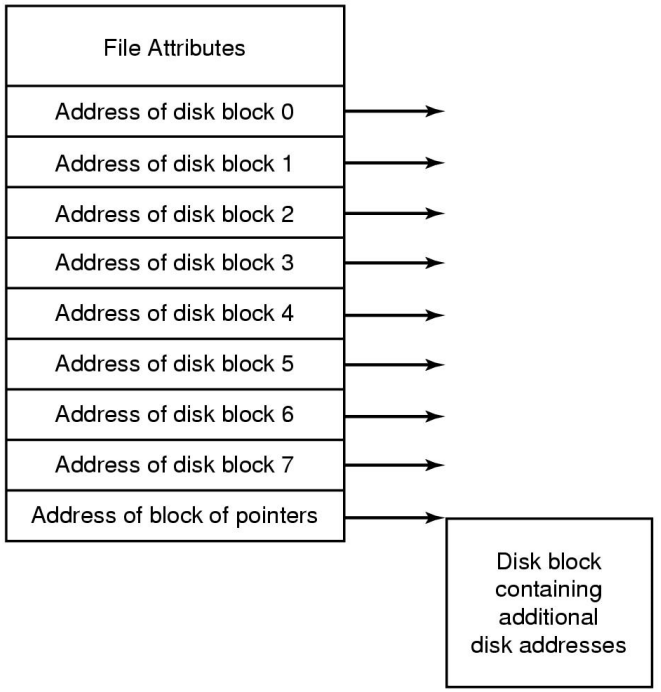
Άσκηση 27

Θεωρήστε τον κόμβο *i* της παρακάτω εικόνας. Αν υποθέσουμε ότι περιέχει 10 άμεσες διευθύνσεις των 4 byte, ενώ όλα τα μπλοκ καταλαμβάνουν 1024 byte, ποιο είναι το μεγαλύτερο αρχείο που μπορεί να υπάρξει;



Απάντηση 27

- Κάθε έμμεσο μπλοκ μπορεί να χωρέσει 1024/4=256 διευθύνσεις δίσκου.
- Μαζί με τις 10 - 1 = 9 άμεσες διευθύνσεις δίσκου, συνολικά 265 μπλοκ.
- Εφόσον κάθε μπλοκ είναι 1 KB, το μέγιστο μέγεθος αρχείου είναι 265 KB.



Άσκηση 28

Μετά από την διαμόρφωση (format) ενός διαμερίσματος δίσκου (disk partition), η αρχή ενός **χάρτη bit ελεύθερου χώρου** είναι η εξής: 1000 0000 0000 0000 (το πρώτο μπλοκ χρησιμοποιείται για το βασικό κατάλογο).

Το σύστημα ψάχνει πάντα για ελεύθερα μπλοκ ξεκινώντας από το μπλοκ με το μικρότερο αριθμό. Επομένως, μετά από την εγγραφή του αρχείου *Α* το οποίο χρησιμοποιεί 6 μπλοκ, ο χάρτης bit γίνεται ως εξής: 1111 1110 0000 0000. Δώστε τη μορφή του χάρτη bit μετά από τις επόμενες ενέργειες:

(α) Γράφεται το αρχείο *Β*, το οποίο χρησιμοποιεί 5 μπλοκ.

(β) Διαγράφεται το αρχείο *Α*.

(γ) Γράφεται το αρχείο *Γ*, το οποίο χρησιμοποιεί 8 μπλοκ,

(δ) Διαγράφεται το αρχείο *Β*.

Απάντηση 28

Αρχική κατάσταση: 1111 1110 0000 0000

(α) Εγγραφή του Β (5 μπλοκ) : 1111 1111 **1111** 0000

(β) Διαγραφή του Α (6 μπλοκ) : 1**000 000**1 1111 0000

(γ) Εγγραφή του Γ (8 μπλοκ) : 1**111 111**1 1111 **11**00

(δ) Διαγραφή του Β (5 μπλοκ) : 1111 111**0 0000** 1100

Άσκηση 29

Θεωρήστε ένα δίσκο με:

- μέσο χρόνο αναζήτησης **8 msec**,
- ρυθμό περιστροφής **15.000 rpm**, και
- **262.144 byte ανά τροχιά**.

Ποιοί είναι οι ρυθμοί μεταφοράς δεδομένων για μεγέθη μπλοκ: (α) **1 KB**, (β) **2 KB**, και (γ) **4 KB** αντίστοιχα;

Απάντηση 29

- Ρυθμός περιστροφής 15.000 rpm: σε 1min γίνονται 15.000 στροφές, άρα η μία περιστροφή σε

  $(60 * 10^3) / 15.000$ msec = 4 msec ανά περιστροφή

- Ο μέσος χρόνος προσπέλασης σε msec για την ανάγνωση ***k* μπλοκ**, θα είναι:

  Χρ. αναζήτησης + χρ. περιστροφής + χρ. μεταφοράς *μπλοκ*

  $$8 + 2 + (k / 262144) \times 4.$$

- Για μπλοκ 1 KB, 2 KB, and 4 KB, ο χρόνος προσπέλασης είναι αντίστοιχα 10,015625 msec, 10,03125 msec, και 10,0625 msec.
- Συνεπώς, ο ρυθμός μεταφοράς δεδομένων είναι περίπου 102,240 KB/sec, 204,162 KB/sec, και 407,056 KB/sec, αντίστοιχα.

Άσκηση 30

Ένα συγκεκριμένο σύστημα αρχείων έχει **μπλοκ δίσκου μεγέθους 2 KB**. Η κεντρική τιμή του μεγέθους αρχείου είναι 1 KB.

Αν όλα τα αρχεία ήταν **ακριβώς 1 KB**, τι ποσοστό χώρου του δίσκου θα πήγαινε χαμένο;

Πιστεύετε ότι οι απώλειες σε ένα πραγματικό σύστημα αρχείων είναι μεγαλύτερες ή μικρότερες από το σύστημα αυτό;

Εξηγήστε την απάντησή σας.

Απάντηση 30

- Για αρχεία του 1 KB, τότε για κάθε μπλοκ των 2 KB, χάνεται χώρος 1 KB (απώλεια χώρου 50%).

- Στην πράξη κάθε σύστημα αρχείων έχει πολλά μικρά αρχεία και μεγάλα αρχεία τα οποία χρησιμοποιούν το χώρο του δίσκου πιο αποδοτικά.

- Παράδειγμα:
  - έστω αρχείο με μέγεθος (1 MB + 1 byte) = $2^{20} + 1 = 1.048.577$ byte.
  - θα χρησιμοποιούσε $(2^{20} / 2^* 2^{10}) + 1 = 2^9 + 1 = 513$ μπλοκ
  - Άρα ποσοστό αξιοποίησης δίσκου:

    $1.048.577 / (513 * (2^*2^{10})) = 1.048.577 / 1.050.624 = 0,99805$

-Τέλος ασκήσεων-