Όνομα: **Valentin Ivanov**
ΑΜ: **1115 2014 00049**

Λειτουργικά Συστήματα

Χειμερινό Εξάμηνο 2016-2017

3ο σετ ασκήσεων

Δώστε σύντομες αλλά περιεκτικές απαντήσεις σε όλα στα παρακάτω ερωτήματα:

- **Πρόβλημα 1 (Πόντοι 30)**

  Ας υποθέσουμε ότι έχουμε το παρακάτω page reference string που εμφανίζεται στην διάρκεια εκτέλεσης ταυτοχρόνων προγραμμάτων:

  $$R = 7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 1, 0, 5, 4, 6, 2, 3, 0, 1$$

  Ας θεωρήσουμε επίσης ότι υπάρχουν 3 frames διαθέσιμα στην κυρίως μνήμη. Βρείτε τα page faults που δημιουργούνται όταν οι παρακάτω πολιτικές ακολουθούνται:

  1. FIFO
  2. LRU
  3. Optimal

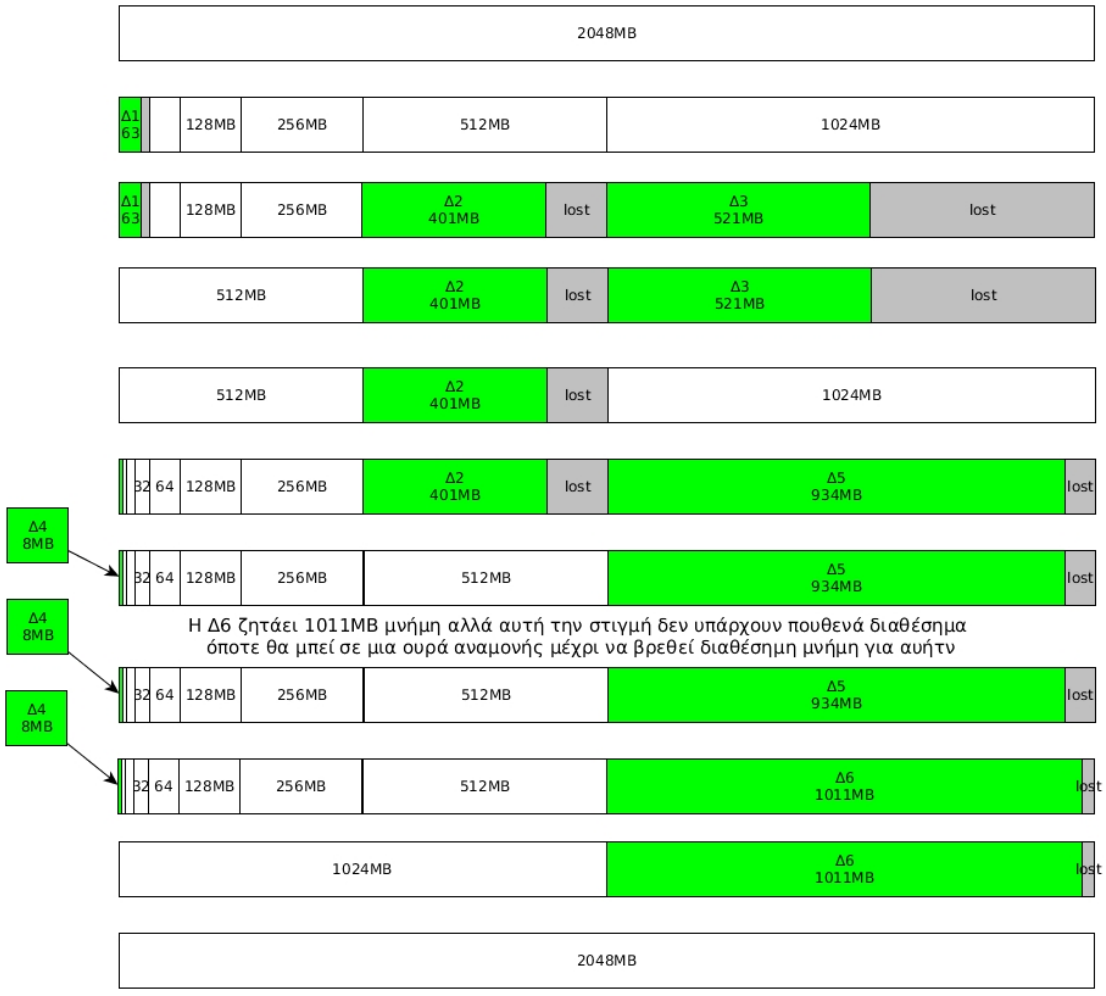
  Δείξτε όλη την δουλειά σας.

- Πρόβλημα 2 (Πόντοι 25):

  Υπάρχει ένα κομμάτι μνήμης που είναι 2GBytes και το οποίο διαχειριζόμαστε με το σύστημα Buddy. Δώστε την κατάσταση της μνήμης όταν έχουμε τα εξής γεγονότα:

  1. Διεργασία Δ1 'ζητά' 63 MBytes την χρονική στιγμή 1.
  2. Διεργασία Δ2 'ζητά' 401 MBytes την χρονική στιγμή 4.
  3. Διεργασία Δ3 'ζητά' 521 MBytes την χρονική στιγμή 10.
  4. Διεργασία Δ1 αποδεσμεύει το χώρο της την στιγμή 13.
  5. Διεργασία Δ3 αποδεσμεύει το χώρο της την στιγμή 22.
  6. Διεργασία Δ4 'ζητά' 8 MBytes την χρονική στιγμή 31.
  7. Διεργασία Δ5 'ζητά' 934 MBytes την χρονική στιγμή 34.
  8. Διεργασία Δ2 αποδεσμεύει το χώρο της την στιγμή 35.
  9. Διεργασία Δ6 'ζητά' 1011 MBytes την χρονική στιγμή 41.
  10. Διεργασία Δ5 αποδεσμεύει το χώρο της την στιγμή 43.
  11. Διεργασία Δ4 αποδεσμεύει το χώρο της την στιγμή 45.
  12. Διεργασία Δ6 αποδεσμεύει το χώρο της την στιγμή 58.

Πρόβλημα 2



- **Πρόβλημα 3 (Πόντοι 15)**

  Υποθέστε ότι έχουμε μια μνήμη με σελιδοποίηση κατ' απαίτηση. Ο πίνακας σελίδων διατηρείται σε καταχωρητές. Η εξυπηρέτηση ενός σφάλματος σελίδας διαρκεί 5ms αν είναι διαθέσιμο ένα κενό πλαίσιο ή αν η σελίδα που αντικαθίσταται δεν είναι τροποποιημένη και 16ms αν είναι τροποποιημένη. Ο χρόνος προσπέλασης της κυρίας μνήμης είναι 80ns.

  Υποθέστε ότι η σελίδα που πρόκειται να αντικατασταθεί είναι τροποποιημένη 85% των φορών. Ποιος είναι ο μέγιστος αποδεκτός ρυθμός σφαλμάτων σελίδας για πραγματικό χρόνο πρόσβασης όχι μεγαλύτερο από 130ns;

Αφού η εξυπηρέτηση ενός σφάλματος θα είναι το 85% τροποποιημένη και 15% ατροποποίητη τότε κατά μέσο όρο η εξυπηρέτηση ενός σφάλματος θα είναι:

$$0.85 * 16ms + 0.15 * 5ms = 14.35ms$$

Ο τύπος για πραγματικό χρόνο πρόσβασης είναι ο εξής

$$efc = (1 - p)80ns + p * 14.35ms$$

όπου p η πιθανότητα να συμβεί σφάλμα. Αρά αφου θέλουμε $efc = 130ns$ θα έχουμε:

$$130ns = 80ns - p * 80ns + p * 14.35ms$$

$$(130 - 80)10^{-9} = (1435 * 10^{-5} - 80 * 10^{-9})p$$

$$p = \frac{0,00000005}{0,01434992} = 0,000003484 = 0,000348434\%$$

Αυτός λοιπόν είναι ο μέγιστος αποδεκτός ρυθμός σφαλμάτων ώστε ο πραγματικός χρόνος πρόσβασης να μην είναι μεγαλύτερος απο $130ns$.

- **Πρόβλημα 4 (Πόντοι 15)**

  Στην συζήτηση στην τάξη για το Working Set Model υποθέσαμε ότι υπάρχει ένα και μόνο έτσι σύνολο για κάθε διεργασία. Είναι πιθανόν καλύτερο/χειρότερο κάθε διεργασία να έχει δύο τέτοια σύνολα; Το ένα μπορεί να αναφέρεται στην ροή εντολών προγράμματος και το δεύτερο στη ροή δεδομένων που το πρόγραμμα χρησιμοποιεί και φυσικά προέρχονται (και αυτά τα δεδομένα) από το δίσκο. Δικαιολογήστε την απάντησή σας.

Απάντηση:
Γνωρίζουμε ότι το WSM διατηρεί τα τελευταία Δ $page\ references$ τα οποία έχει χρησιμοποιήσει το πρόγραμμα μας. Σε αυτά τα Δ το πιο πιθανόν είναι να βρίσκονται και σελίδες που περιέχουν δεδομένα. Το Δ έχει συνήθως τιμή 10-15 χιλίαδες και πρέπει να είναι άμεσα και γρήγορα προσβάσιμο στην $cache$ ή ακόμα και να αποθηκευτεί στον $translation\ lookaside\ buffer\ (TLB)$.
Αν είχαμε κι άλλο ξεχωριστό $Working\ Set$ για τα δεδομένα τότε θα αυξανώταν η πολυπλοκότητα στους υπολογισμούς και τους ελέγχους. Πιθανόν και να αυξανώταν το μέγεθος των πληροφοριών που κρατάνε τα $working\ sets$ όποτε κάθε φορά που μια διεργασία θα γινόταν $swap\ in - out$ θα έπρεπε να μεταφέρονται παραπάνω σελίδες, με αποτέλεσμα να οδηγηθούμε ακόμα και στο φαινόμενο $TLB\ thrashing$.

- **Πρόβλημα 5 (Πόντοι 15)**

  Θεωρήστε ένα υπολογιστικό σύστημα που παρουσιάζει τα ακόλουθα ποσοστά χρήσης (utilization) σε διάφορες devices:
  - CPU utilization: 20%
  - Χρήση δίσκου σελιδοποίησης: 97,9%
  - Χρήση κυρίας μνήμης: 42%
  - Άλλες συσκευές I/O: 5%

  Για κάθε ένα από τα ακόλουθα σενάρια εξηγήστε αν είναι πιθανόν να βελτιωθεί η κατάσταση στο εν λόγω υπολογιστικό σύστημα.
  1. Αγορά πιο γρήγορης CPU?
  2. Εγκατάσταση μεγαλύτερου δίσκου (σε χωρητικότητά);
  3. Αύξηση του βαθμού πολυπρογραμματισμού;
  4. Αύξηση μεγέθους μνήμης;
  5. Αγορά πιο γρήγορης μονάδας για 'άλλα I/O';
  6. Εγκατάστασή δίσκου που διαθέτει πιο πολλές περιστροφές και πιο γρήγορη ταχύτητα μεταφοράς δεδομένων;
  7. Αύξηση του μεγέθους της σελίδας;
  8. Μείωση του βαθμού πολυπρογραμματισμού;

Απάντηση:

Το *utilization* ενός *device* είναι το ποσοστό του χρόνου στο οποίο αυτό κάνει χρήσιμη δουλεία για τον χρήστη. Αν για παράδειγμα η CPU το 90% του χρόνου κάνει δικούς της υπολογισμούς και εργασίες τότε αυτό που μένει για δουλεία του χρήστη είναι μονο το 10%. Έτσι το *utilization* είναι 10%.

Γενικά θέλουμε τα *utilization* των διαφόρων συκευών να είναι μεγάλα ποσοστά ώστε να παράγουν όσο περισσοτερο χρήσιμη δουλεία πμορούν.

Για τις παρακάτω συσκευές έχουμε:

1. Αφού η CPU έχει *utilization* μόνο 20% τότε αυτή αφιερώνει πολύ χρόνο στο να κάνει άλλα πράγματα αρά παράγει λιγότερο χρήσιμη δουλεία για τον χρήστη.
Αν αγοράσουμε πιο γρήγορη CPU τότε σε οποιαδήποτε περίπτωση θα έχουμε βελτίωση στην ταχύτητα του υπολογιστή.

2.Ο μεγαλύτερος σκληρός δίσκος μας εξασφαλίζει περισσότερο χώρο για σελιδοποίηση-*swapping*. Βλέπουμε ότι σχεδόν όλος ο χώρος σελιδοποίησης χρησιμοποιείται αρά δεν φτάνει η κυρία μνήμη. Με ένα μεγαλύτερο σκληρό δίσκο θα μπορούμε να εξασφαλίσουμε ότι δεν θα γεμίσει ποτέ ο χώρος σελιδοποίης, πράγμα που πιθανόν αν δημιουργούσε προβλήματα.

3.Αν αυξηθεί ο βαθμός πολυπρογραμμάτισμού τότε θα έχουμε περισσότερες διεργασίες που να τρέχουν ταυτόχρονα. Αν όμως όλες αυτές οι διεργασίες χρειάζονται πολύ κυρία μνήμη RAM τότε αυτό θα οδηγήσει σε ανεπάρκεια αυτής και σε επιβάρυνση του συστήματος λόγο $page\ faults$ και $swapping$που θα γίνονται. Αρά δεν θα είναι καλή ιδεά σε αυτή την κατάσταση να αυξήσουμε τον βαθμό του πολυπρογραμματισμού.

4.Αυξάνωντας την κυρία μνήμη είναι πάντα καλή ιδέα εφόσον το υποστηρίζει το υλικό. Με μεγαλύτερη κυρία μνήμη η CPU δεν θα επιβραδύνεται με *swapping* όποτε θα έχουμε καλύτερο *utilization* εφόσον θα χωράνε περισσότερες σελίδες *pages* στην μνήμη αρα λιγότερα $page\ faults$.

5.Εφόσον έχουμε μικρό *utilization* των συσκευών E/E με γρηγορότερες συγκεύες θα βλέπαμε καλύτερη απόδοση.

6.Ένας γρηγορότερος δίσκος θα έκανε το φόρτωμα των προγραμμάτων όπως και το *paging* πίο γρήγορο. Έτσι θα είχαμε βελτίωση στην συνολική ταχύτητα.

7.Με αύξηση μεγέθους της σελίδας θα χωράνε περισσότερα δεδομένα σε αυτήν. Επίσης

η μεταφορά της σελίδας απο τον σκληρό δισκο στην μνήμη θα αυξηθεί. Υπάρχουν πολλά μικρά προγράμματα που δέν χρειάζονται ολόκληρη την σελίδα. Έτσι με την αύξση του μέγεθός της θα αυξηθεί και ο βαθμός του $fragmentation$ που υπάρχει. Παλίοτερα με την πρόοδο της τεχνολογίας το μέγεθος της σελίδας αυξανώταν. Στα περισσότερα σημερινά συστήματα προσωπικής χρήσης έχει καθιερωθεί το μέγεθος της σελίδας να έιναι 2, 4 ή 8KB και χρειάζεται περεταίρω αύξηση.
8.Με την μείωση του πολυπρογραμματισμού θα έχουμε λιγότερες διεργασίες να διαμοιράζονται την κυρία μνήμη. Έτσι πιθανόν θα μειωθεί η ανάγκη του $swapping$ και θα έχουμε καλύτερη απόδοση στα λιγότερα προγράμματα που θα τρέξουμε. Αρα έχουμε βελτίωση.